SUPPORT FOR LOCATION AND COMPREHENSION OF USER HISTORY IN COLLABORATIVE WORK

A Dissertation

by

DO HYOUNG KIM

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2011

Major Subject: Computer Science

Support for Location and Comprehension of User History in Collaborative Work

SUPPORT FOR LOCATION AND COMPREHENSION OF USER HISTORY IN COLLABORATIVE WORK

A Dissertation

by

DO HYOUNG KIM

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee, Frank M. Shipman, III
Committee Members, Richard K. Furuta
James Caverlee
Ergun Akleman
Head of Department, Duncan Walker

December 2011

Major Subject: Computer Science

ABSTRACT

Support for Location and Comprehension of

User History in Collaborative Work. (December 2011)

Do Hyoung Kim, B.S., Soongsil University;

M.S., Seoul National University

Chair of Advisory Committee: Dr. Frank M. Shipman, III

Users are being embraced as partners in developing computer services in many current computer supported cooperative work systems. Many web-based applications, including collaborative authoring tools like wikis, place users into collaborations with unknown and distant partners. Individual participants in such environments need to identify and understand others’ contributions for collaboration to succeed and be efficient. One approach to supporting such understanding is to record user activity for later access. Issues with this approach include difficulties in locating activity of interest in large tasks and the history is often recorded at a system-activity level instead of at a human-activity level. To address these issues, this dissertation introduces CoActIVE, an application-independent history mechanism that clusters records of user activity and extracts keywords in an attempt to provide a human-level representation of history. CoActIVE is integrated in three different software applications to show its applicability and validity. Multiple visualization techniques based on this processing are compared in their ability to improve users' location and comprehension of the activity of others. The results show that filmstrip visualization and visual summarization of user activity show significant improvement over traditional list view interfaces.

CoActIVE generates an interpretation of large-scale interaction history and provides the interpretation thorough a variety of visualizations that allow users to navigate the evolution of collaborative work. It supports branching history, with the understanding that asynchronous authoring and design tasks often involve the parallel development of alternatives. Additionally, CoActIVE has the potential to be integrated into a variety of applications with little adjustment for compatibility. Especially, the comparison of visualizations for locating and comprehending the work of others is unique.

DEDICATION

To my father and mother

ACKNOWLEDGEMENTS

It has been a long journey, but I am very happy to share good news with those who have encouraged me to achieve my goal. Most of all, I would like to thank my wife, Hyoeun. Whenever I faced obstacles in my work, she was always there to help me. My precious Claire Rina made my life full of pleasure. Her bright smile made me forget all my troubles. My parents have always been reliable supporters. Even when my studies had been delayed, they still supported me and cheered me on during the journey.

Furthermore, I want to express my appreciation for my advisor, Dr. Frank M. Shipman III. Without his guidance, this long journey wouldn't have been successful. He was an admirable scholar as well as an excellent mentor to me. I also want to give my thanks to Dr. Richard K. Furuta, Dr. James Caverlee, and Dr. Ergun Akleman, for their feedback and review on my research.

I also want to thank my friends and colleagues. Particularly, Soonil, Haowei, and Michael, who were great help to me during my graduate studies. Additionally, I will never forget my friends and lab members: Anna, Arun, Chris (who was my housemate), Konstantinos, Prasanth, and Suinn.

Lastly, I give all the glory to God for all that he has done in my graduate life at Texas A&M University.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

With the increase in server-based and cloud-based computing, there has been a corresponding increase in applications supporting remote collaborative work. For example, wikis give users permission to add, edit, and remove contents to promote their collaboration, information sharing, and communication. This means that collaborations are moving from being between people that know one another to potentially anonymous collaborators with little shared understanding.

However, without sufficient understanding between collaborators, such remote collaboration cannot be efficient and productive. For example, individuals may duplicate work already done by others or may unknowingly delete others' work while instantiating their work. Such situations can arise in most collaborative applications, such as turn-taking document editing where authors invalidate other collaborators' edits. Replicated effort and destructive editing are appropriate in many situations but a lack of understanding between remote collaborators increases the likelihood of accidental and unnecessary occurrences.

For effective collaboration, people need to understand the efforts and motivations of others. As for understanding the efforts and motivations of co-workers, one approach is to record user activity for later access. By following the activities of collaborators, a

---

This dissertation follows the style of the *ACM Transactions on Information Systems*.

user can have insight into their thought processes [Bush 1945]. Tools that allow users to follow the steps of other collaborators' work can help users resolve ambiguities caused by (1) the diverse backgrounds of users, (2) the complexity of their work, and (3) interaction barriers [Lee 1992]. Through access to history of activities, the participant can not only understand the intentions of their collaborators, but also increase the value of their contribution based on an improved understanding of the past work history.

Despite the benefit of automatically preserving records of user activity [Lee 1992; Reeves 1993] , there are challenges for its presentation and use in a system. First, locating activity of interest in large tasks becomes a potentially time-consuming activity. Second, users can have problems comprehending automatically-recorded history, which is represented at a system-activity level (e.g. transaction records) instead of at a human-activity level. Branching history, which is used to represent alternative directions of problem solving or design, results in users having difficulty in navigating between related portions of activity in the history and understanding how different branches are related. These problems are generally independent of the domain of the collaboration and motivate the research on representations and interfaces for user history.

This dissertation introduces CoActIVE, a history mechanism designed to improve users' location and comprehension of collaborative work. The mechanism clusters history records and extracts keywords in an attempt to provide a more human-level representation of history. Based on this processing, it provides multiple visualizations of collaboration history. CoActIVE is developed as a Java library, which can be integrated in Java applications.

Evaluation was performed on CoActIVE integrated in the Visual Knowledge Builder (VKB), a publicly-available Java application that includes history recording [Shipman et al. 2001]. Four VKB documents collaboratively authored by students in remote locations were given to participants. Each participant was asked to fulfill four tasks using diverse history visualization components. Those components were compared in their ability to improve users' location and comprehension of the activity of others. The results show the new visualizations result in a significant improvement over prior interfaces.

The next chapter describes problem and issues surrounding the role of history in collaborative environments. Chapter III introduces related work in five parts: navigation and comprehension of history, history visualization, awareness, branching history, and undo and redo. After that, Chapter IV presents the approach of this research.

Chapter V presents the design and development of CoActIVE, and Chapter VI is the detail of the CoActIVE library. Chapter VII describes the integration of the library into VKB, the Design Exploration (DE) software design tools [Moore 2007], and VKB Server. These systems already have mechanisms for collecting user history yet their history-based functionality is limited due to the problems previously described. Chapter VIII reports on an evaluation comparing four history interfaces based on the techniques described in this dissertation, and Chapter IX concludes with the contribution of this dissertation and presents future work.

CHAPTER II

PROBLEM AND ISSUES

A variety of systems collect records of user activity as history. Such systems often enable users to navigate through the history to see previous work by going back and forth between the current and prior state [Shipman and Hsieh 2000]. For a single-person task, this acts as a reminder of prior activity and decisions, while in collaborative applications it provides a source of awareness and understanding of the work of others. As described in the context of hypertext navigation by Bush, following the trails of other users' activities provides insight into their thought processes [Bush 1945]. For group projects with changing group membership, navigating the efforts of collaborators can give a new participant hints to understand group work more quickly and provide insight into the decisions of those no longer participating in the project.

2.1 Understanding of System-Level History

A variety of systems collect records of user history enabling users to review their own work or to understand other collaborators' work. The typical method for representing user history is to record edit events to the underlying data structures involved in an application as user activity occurs in the interface. The resulting record includes low-level properties such as event type and timestamp and is a slight extension over the representation needed for undo/redo capabilities.

However, the fine-grained representation of events can create problems for user interaction with history due to a mismatch with user's understanding of their activity.

For example, in a document editing application, an operation of inserting a figure in the middle of a page would be recognized as single activity to human users. But, at the low-level of the application, this activity can be accomplished by applying a series of system-level events such as “import a figure”, “move a figure”, “resize a figure”, and so on. Moreover, individual user actions make up activities.

Such a difference was described by Rosenberg, in the context of reading hypertexts as the collection of low-level actions he calls actemes (such as the action of following a link) into coherent episodes (such as the exploration of a particular idea) [Rosenberg 1996]. A few systems address this mismatch by allowing users to manually group the system-level history events into human-level activity [Shipman, Hsieh, Maloor and Moore 2001], but this grouping task is not part of users’ main goals so is unlikely to happen in real settings and becomes unmanageable in terms of the effort required as history grows. This problem becomes more severe in collaborative settings since users are often left to interpret collaborators’ intention by recognizing coherent activities from the recorded history.

2.2 Navigation and Orientation Difficulties in History

As collaborative work continues, the activity history can quickly become very large increasing orientation and navigation difficulties. When users navigate the history, they may want to find a specific activity and understand it. For one’s own efforts, users may remember enough about the order of their activities to locate the particular activity in question. However, for efforts that extend over long period the increasing size of the history and the dimming of the user’s memory can result in difficulty. When the activity being located was performed by someone else, the user has less understanding to orient themselves in the overall activity relative to the activity in question.

Most history interfaces, e.g. controls for playing forward and backward, lists and even scrubbing sliders, increase the time and effort required for locating a particular activity in the history as the history increases in size. Once found, users can be disoriented when they navigate between different points in a history without an indication of how far in time and effort they have traveled.

In general, there is often a trade-off between the effort required to locate an activity and the orientation provided by the interface. Interfaces that provide more direct access to prior activities provide less feedback on the intermediate activities being skipped. Consequently, techniques are necessary to facilitate users' understanding of where they are in a recorded history and where particular sub-activities occur within that history.

2.3 Branching History

Since diverse solutions exist for many problems, users are likely to try alternative work paths during collaboration [Terry et al. 2004]. Branching history reduces the risk of trying out alternative solutions by representing each solution as a branch in the history. This allows the users to explore diverse solutions without overwriting the work of others, making such explorations more socially acceptable.

Through the navigation of branching history, users can examine the alternatives developed by their collaborators to get ideas. They can also make their solutions from related activity instead of starting the problem-solving process from the beginning.

Branching history has become part of single-user applications where alternate series of modifications need to be compared, including Adobe Photoshop.

The structure of branching history aggravates user navigation and comprehension of history. While users are navigating through multiple history branches that happened concurrently, they can face difficulty due to the complex structure of the branching history. The history's complex structure can require users to understand not only where they are in the activity but with which branch of the activity they are currently viewing, how that branch is related to other branches, and who is involved in that branch.

This difficulty is greater in on-line collaborative work settings because users do not know what other work is progressing at the same time and who is involved in the work. Also they cannot rely on their memory to infer such information. When such information is not available, users tend to repeatedly navigate all alternative solutions in turns [Derthick and Roth 2000]. This is a problem when the number of alternatives is increased.

2.4 Complexity of Implementation

History has a great potential to support collaboration and will almost certainly become part of more applications. Even though many of the issues concerning history comprehension and navigation are independent of the application, most applications provide relatively simple history mechanisms. The history flow visualization is a good example. It was developed to understand the collaborative dynamics which are found in a wiki's revision history. But through its visualization, users cannot make sense of the available version history of a wiki document [Vigas et al. 2004]. Any implementation that provides insufficient support for understanding history is likely to cause adverse reactions such as repetitive undo/redo and user disorientation.

CHAPTER III

RELATED WORK

The research into support for location and comprehension of user history in collaborative work has five areas of related work: navigation and comprehension of history, history visualization, awareness, branching history, and undo/redo operations in collaborative settings.

3.1 Navigation and Comprehension of History

History information has been employed for various purposes such as reuse, navigation, reminding, error recovery, user modeling, user interface adaptation, and inter-referential I/O [Lee 1992]. While some of these uses are for single-user systems, history information has great potential to facilitate collaborative work by increasing awareness and understanding between users.

When people are working together in a project, the results of collaborators' past activities can be ambiguous. For example, there may be multiple potential reasons to place elements near one another in a design meaning that collaborators are unaware of what was the thought process behind particular design decisions. To compensate for this ambiguity, Reeves' collaborative design system, INDY, enabled users to replay and trace collaboration history as well as return to specific events in history. By observing the activity near the particular action in question, users were more able to resolve ambiguities in understanding the progress and meaning of collaborative work [Reeves 1993].

VKB includes a history mechanism similar to that in INDY, including a variety of methods for navigating the history. It provides four interfaces; history play buttons to play history forward and backward, a history slider for random access of history, a history session dialog that presents a list of history events, and a popup menu interface to return to events concerning specific elements in a workspace. The history session dialog allows users to group system-level history events into higher-level activities manually [Shipman, Hsieh, Maloor and Moore 2001].

Although VKB's history mechanism is helpful to access history, navigation within large history limits its effectiveness. For example, the position of the slider and the timestamp are only valuable if users have some memory or understanding of what activities were happening when. Such memory fades over time and may not be available at all in collaborative settings. Similarly, the history player can provide an overview of activity via the animation of prior activity but takes too long in large histories.

Automatically grouping events was explored in SmartBack, which identified important states in a history of web navigation to provide more direct access to the most valuable pages in a history [Milic-Frayling et al. 2004]. In a user study of SmartBack, users reported a qualitative improvement in the browsing experience.

This result points towards the potential value of grouping the system-level events in a history record. For grouping edit events, applications have included a simple time gap approach to define sessions or noted application start times to generate checkpoints in the history of an artifact. With the idea that fixed time gaps would not work for all activities, Shirai et al. [2009] developed a time-slicing method to group events for history summarization.

3.2 History Visualization

Numerous studies visualize aggregated history information to enhance user comprehension of collaborative work. Hill and Hollan's edit wear and read wear visualize the history of author and reader interaction with documents onto document scrollbars [Hill et al. 1992]. This visualization allowed users to identify useful features of documents, such as the slowest changing sections and more highly read sections of a document.

Plaisant et al. [1996] propose LifeLines that visualizes summaries of personal history. It provides an overview of a personal history with visual cues, such as line color and thickness. Users can also filter the history to focus on part of the record in detail. This technique was also employed in an educational setting [Plaisant et al. 1999]. To facilitate collaborative learning, their history mechanism allows students to review the records of each other's work through the visualization of their activity along with the progress of simulation.

Similarly, Begole et al. [2002] visualize the history of user activity when using a computer and accessing e-mails to extract meaningful patterns of common activity between individuals and within individuals according to time of day, location, and day of the week. Viégas and colleagues introduce a history flow tool to make broad trends in revision histories immediately visible, while preserving details for closer examination. This tool was designed to visualize relationships between multiple document versions in the Wikipedia corpus. From the visualization it revealed the patterns of cooperation and conflict among those versions [Vigas, Wattenberg and Dave 2004].

The goals of such systems are to provide information about general trends in activity rather than supporting the location and comprehension of specific portions of a record of events.

Closest to the visualization approach of our work, Nakamura and Igarashi [2008] employs a comic strip metaphor to visualize user activity which occurred in Java GUI applications. To summarize the history visually, they provide visual cues, such as word balloons for keyboard operations as annotations on each snapshot.

3.3 Awareness

In collaborative work environments, information concerning each user's activity can be visualized in diverse ways. Visualizations of the activity of others for awareness purposes is often part of synchronous collaborative applications, so users have a sense for what their geographically distributed co-workers are doing.

Zellweger et al. [2003] use a "city lights" metaphor to provide information about unseen objects. Just as the lights from a city are visible from a distance at night as a glow on the horizon, they utilize window borders to indicate the existence of the veiled objects. On the other hand, Baudisch and Rosenholtz [2003] employ a "streetlamps" metaphor to support spatial cognition of an off-screen object by visualizing rings that surround the object and are just large enough to reach into the border region of the display window. Both techniques provide solutions to the problem of indicating activity that would otherwise not be visible due to the user's current view of a shared artifact. A similar problem occurs during the playback of grouped history events as multiple activities may be simultaneously presented and the system must decide which to present through animation and which to present through other visual cues.

3.4 Branching History

Branching history poses challenges for navigation and has been a focus of history visualization. The Designers' Outpost system's history interface employs filmstrip visualization technique to provide the sense of work process. It supports branching history by presenting only the current working history linearly with collapsed stubs which users may select to navigate to another history branch [Klemmer et al. 2002].

The Parallel Paths also allows users to create and compare the alternative paths of image manipulation task simultaneously in the same workspace [Terry, Mynatt, Nakakoji and Yamamoto 2004]. In the WWW environment, Footprints provides a site map and a path map to provide past users' navigation history [Wexelblat and Maes 1999].

Besides, the management of branching history has been widely conducted in various fields of computer science such as software engineering [Nguyen et al. 2004] and data analysis [Derthick and Roth 2000]. However, most research does not present how to manage branching history information efficiently, limiting their scope to application-specific support. Assuming that collaborative work goes on for long period of time and the history of the collaboration contains thousands of edits over periods of years, it is necessary to consider efficiency in the representation and management of branching history.

3.5 Undo and Redo

Supporting undo/redo in synchronous collaborative work is an active area of research. Chengzheng [2000] presents a consistent undoable operation regardless of its undo context. Through operational transformation, the undo operation can be interpreted as a concurrent inverse operation. Berlage [1994] provides an undo operation for the application with a graphical user interface by executing the inverse of the operation.

CHAPTER IV

APPROACH

The prior chapters have described the potential of history information to facilitate the understanding of collaborative work and prior research on how to provide the information properly. Building on this prior work, the emphasis of this dissertation is to investigate (1) how to resolve the problems of the mismatch between the level of human activity and that of system representation, and (2) how to address the navigation and orientation difficulties in the history of collaborative work. To help ensure the broad applicability of the approaches described, this dissertation also examines the more practical issues of ensuring the approach supports branching history and can work with a variety of collaborative applications.

To address the issues previously described, our approach is to develop a history mechanism that aggregates the system-level record of history into higher-level and hopefully more meaningful activities and visualizes salient characteristics of these activities.

4.1 Meaningful Aggregations of System-Level History

The set of low-level events recorded by a system during user activity are a partial record of the human effort. Human activities are often described, whether retrospectively explaining ones' actions or in planning for future action, via a vocabulary of tasks and sub-tasks [Suchman 1987]. The goal of aggregating the system-level activity is to form a hierarchic representation of the recorded system-level events that roughly corresponds to the tasks and sub-tasks people would use to describe the activity.

Such an aggregation would help users more quickly comprehend the relationship between the history and their memory/understanding of the activity. A multi-level representation would benefit larger tasks if the levels correspond to the participants' mental model of the tasks and sub-tasks of their activity.

To generate such a representation, we employ Hierarchical Agglomerative Clustering (HAC). This algorithm repeatedly clusters history events until a list of low-level events is organized as a tree representing a hierarchic structure of human activity. HAC relies on a distance function in order to determine which low-level events and previously-generated groupings to merge next. While others have explored purely time-based clustering [Shirai, Yamamoto and Nakakoji 2009], i.e. merge the next pair of subclusters with the smallest gap, such a function will not work in a synchronous CSCW environment where there are expected to be activities corresponding to separate simultaneous sequences of edits. While simply using time-based clustering for each individual user solves most of such problems, it will misrepresent the rapid transition between activities. As a result, we enable an application-specific distance model that can be used to include numerous features of the system-level record of activity (e.g. time, user, content, location, etc.).

4.2 Navigation Interfaces and Visualizations

In our work, history's value lies in its ability to provide insight to users. Such insight requires that users are able to locate and comprehend periods of activity relevant to their current work. Thus, navigation interfaces and visualizations are central to such a process.

The navigation interfaces need to facilitate users' understanding of specific states of the history. The history states are characterized by their point in time and by the representation of the domain-specific artifact at that time. By displaying lists of times and an indication of the associated artifact states, users get a sense for the flow of activity. Expanding from a list to a tree visualization facilitates a hierarchical structure that indicates the relation between states via sequence of tree nodes, levels of the tree, etc. but requires user interaction to open/close branches of the tree to show/hide details.

Another goal of these interfaces is to help users understand the process of work that led to particular states. Processes are composed of series of actions. While static visualizations, especially filmstrip visualizations of artifacts, provide some sense of process, animations that play back user activity provide a natural view of process.

When history becomes large, visualizations that represent the whole structure of the history in detail are not possible. Instead, the visualization and interface must select specific points in the history to present or determine abstract characteristics of the history states or event sequences which will provide adequate information to the user.

Our approach is to provide navigation interfaces that visualize the aggregated history. Through the interfaces users will be aware of their current location in the history and be able to navigate to other locations in the history. We employ techniques to generate visual/textual hints to better represent the history segments resulting from aggregation. Simultaneously, the visualizations provide opportunities for users to access event-level information when needed.

The novel approaches to aggregating and visualizing history can be combined with prior techniques for making history manageable and useful. For example, history playback interfaces and history filters can be employed to improve users' navigation and comprehension of history. Playback interfaces can be utilized when reviewing event sequences and accessing history states selected from the aggregated history. History filters can be used to investigate the history of a certain member's work by presenting only that member's work in the history interfaces.

4.3 Branching History

Our approach includes the support of branching history to help users construct and navigate interleaved works that happened alternatively during collaboration. By visualizing how each branch is related to the others, the interfaces, such as a tree view, provide not only which branch of the activity they are currently viewing and how that branch is related to other branches.

When a new branch is created, though branching action logically occurs in the past, chronologically it occurs in the present. We have explored representations that efficiently store and provide access to the logical structure of branching history.

4.4 Application-Independent History Mechanism

Many applications would benefit from support for activity location and comprehension and already include basic undo/redo capabilities. By designing the instantiation of the above approach to rely on such existing features of applications, we show the potential for application-independent history mechanism. In particular, the next chapter describes CoActIVE, which is short for the Collaborative Activity Interpretation and Visualization Engine. CoActIVE can be incorporated in existing JAVA applications and only requires temporal and action information but allows developers to include application-specific features (e.g. an application-specific distance function for clustering events) as desired.

CHAPTER V

CoActIVE: A NEW HISTORY MECHANISM

CoActIVE (the Collaborative Activity Interpretation and Visualization Engine), is a Java library that can be integrated into Java applications that already have an undo/redo capability. It attempts to address issues with navigation and comprehension found in prior history mechanisms through the automatic aggregation of system-level history and multiple visualizations.

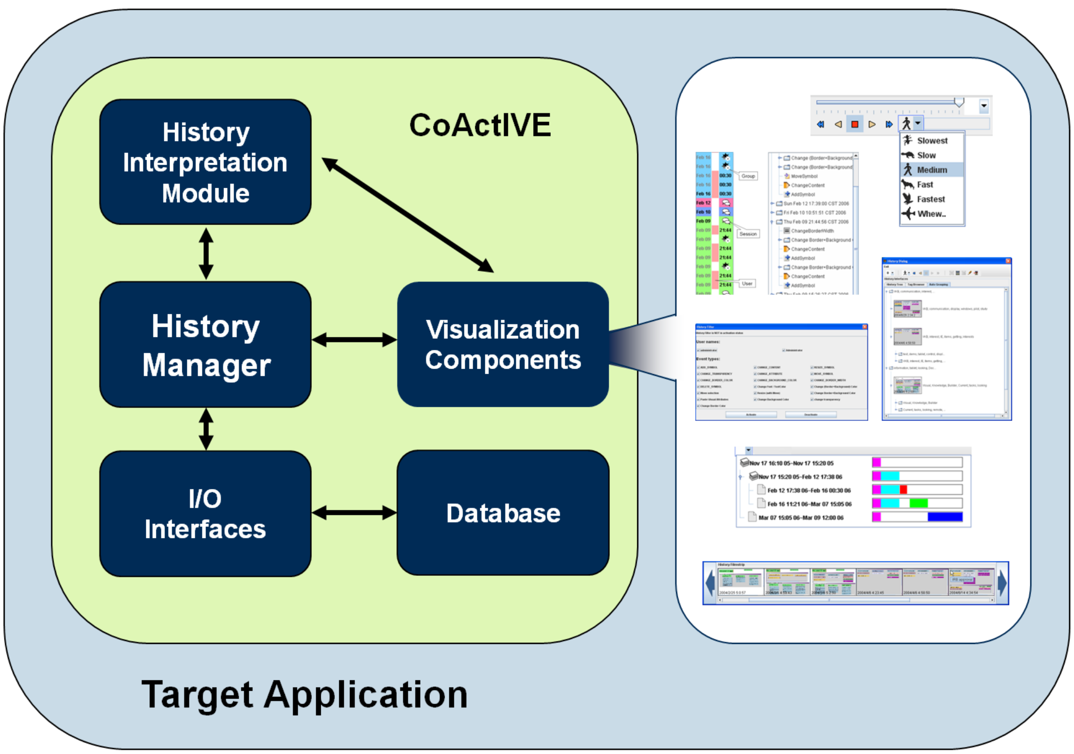


Figure 1. CoActIVE architecture

Figure 1 shows the CoActIVE architecture. It consists of five main parts: a history manager, a history interpretation module, visualization components, I/O interfaces, and a database.

5.1 History Manager

The history manager is the main part of the history mechanism. It creates a history event when a user activity occurs, manages it in an underlying structure, and provides history information to the other parts.

User activity in an application is represented as a system-level event or a series of events by the history manager. When those events occur, the manager stores them in an event list, and manages them in the list chronologically. Through the access of the list, the history manager can perform undo and redo operations of recorded user activity. This list is stored in an internal database permanently for future access, and the manager uses I/O interfaces to access the database transparently. The recorded history is provided to the history interpretation module and visualization components for additional processing.

System-level event structures include the information required by the history mechanism. The information stored for each event includes an event id, an event type, a task description, a username, and a timestamp. The event id is unique and assigned by the history manager.

Consecutive events executed to achieve a particular activity can be recorded as a single batch event. Many applications such as Photoshop Actions [photoshop] support recording and performing a sequence of events to improve workflow, and the history manager supports it via the batch event. The manager assigns a unique id to the batch event to manage it as a single event.

Each application will have its own set of event types – these generally correspond to the alternatives actions found in the application's existing undo/redo software. Event types are required to be defined when CoActIVE is integrated into an application although the set of event types can be expanded later as the application is revised. If necessary, information specific to each event type can be attached to the event to aid in later playback and visualization.

5.1.1 Basic History Representation

The list of history events in the history manager is used to support all history interaction and visualization. Most importantly, the history representation must include enough information so that the manager can rebuild any previous state in the time line of the history. The information in the representation must enable undo and redo. The undo operation means that a user goes back to the past time by one event. If the user performs an undo operation repeatedly, the user can observe the change of his/her work in reverse time order. The redo operation enables the user to navigate his/her work state in forward time order.

To maintain the information of where a user is in the timeline during history navigation, the history manager employs a "navigation id" internally. In most case, the navigation id indicates the id of the most recent event because users usually create new events while working on an application. But, when the users navigate back in the timeline, the history manager continuously performs an undo operation and updates the navigation id with the id of a just previous event of an “undone” event. Similarly, when a redo operation is performed, the navigation id is the id of the event of the operation.

5.1.2 Representing Branching History

While branching history is not new, many history mechanisms – such as MS Word’s revision history – do not support branching history but allow only the linear navigation of prior activity. Therefore, users cannot make modifications at earlier times without losing all events subsequent to that state in the history.

CoActIVE’s history manager maintains branching history to represent alternative paths of user activity. It represents the branching history via history segments, which enables efficient interaction with the branching history. This section describes how to manage multiple versions of collaborative work by splitting its branching history into segments.

A history segment is a chunk of temporally-continuous history events and it is a basic unit of a history segment table. Branched history is represented as graph of history segments in order to reduce the history manager’s memory requirements, which can be large for long histories.

A segment is formed when a history event occurs that is not logically in order with the previous chronological event. There are two ways for this to occur. An event that occurs at a prior work state results in a new branch. For example, in Figure 2, the 6th event (e6) causes a new segment (segment 2) to be created to represent the new branch and divides the existing segment (segment 1) into two segments (segments 1’ and 1’’). Segments are also created for continuations of an existing branch. For example, the

10th event (e10) in Figure 2 causes the formation of segment 3 as the author came back to the end of the first version of the document (segment 1'') after editing the version at segment 2.

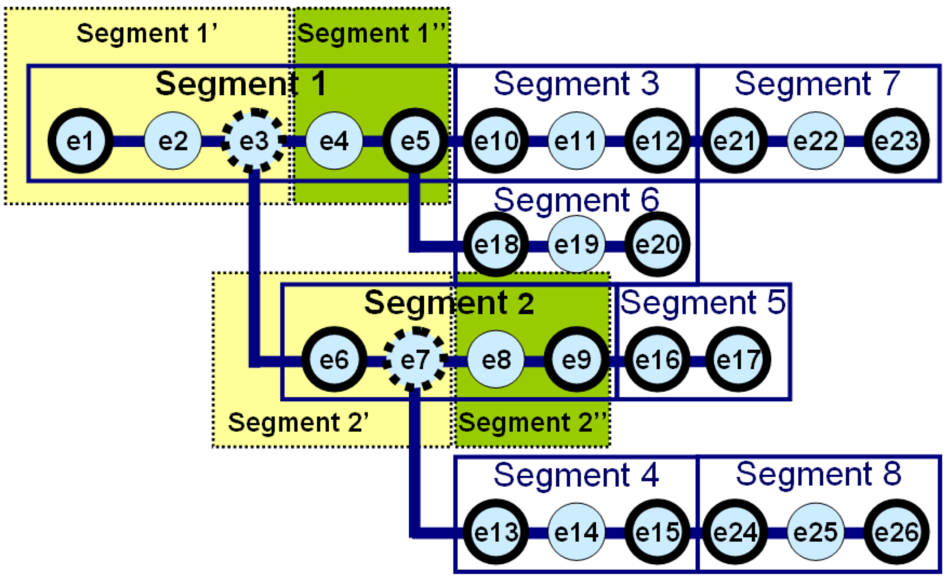


Figure 2. Branching history with segmentation

The segment table maintains the branching history that is used to build and track different versions of work. The segment table in Table I shows the segment table constructed for the branching history in Figure 2. In the table, "Previous Segment" identifies the logically previous segment for the new segment. When an event branches in the middle of an existing segment, that segment is split into two segments. "Start / End Event" sections indicate the first and the last event of a segment. Therefore, the segment table must be updated whenever a history event occurs.

Table I. Segment table

Segment Table

Segment	Previous Segment	Start Event	End Event
1 → 1’	Root	1	5 → 3
1”	1’	4	5
2 → 2’	1’	6	9 → 7
3	1”	10	12
2”	2’	8	9
4	2’	13	15
5	2”	16	17
6	1”	18	20
7	3	21	23
8	4	24	26

(→ : update)

Since the CoActIVE history manager supports branching history internally, an application developer does not need to implement it themselves. When a new event occurs, the manager determines if it is just an addition to the current history branch or if it is issued within another branch or at the previous time in a branch causing new segments to be created.

5.2 History Interpretation Module

The events that make up history records tend to be at a system-level (e.g. low-level edit events and database transactions) rather than at a human-level. As previously described, this means that examining the content of such a record is likely too labor intensive and tedious to be reasonable for many users.

To support users in understanding large quantities of work history quickly, CoActIVE's history interpretation module generates a higher level representation of the history in order to aid the user. It clusters the individual history events into (hopefully) meaningful groups and generates summaries for these clusters. This relies on having an appropriate clustering approach. With the appropriate clustering, users can match the groups in history to human-level activity.

5.2.1 History Clustering

Conceptually, the goal of history clustering is for the tree to represent a form of hierarchic task decomposition of the effort that took place in the history. The interpretation module uses Hierarchical Agglomerative Clustering (HAC) to group similar events together into an initially unknown number of higher-level activities in a two-step process.

Figure 3 illustrates the two-step process of HAC approach: first clustering then thresholding. First, the interpretation module computes the distances between each pair of consecutive events. In the module, each history event is initially regarded as a single leaf node, and similar nodes will be grouped until the remaining nodes are too far apart to be viewed as part of the same activity. Therefore, it is important to define a distance function that determines what it means for events to be similar.

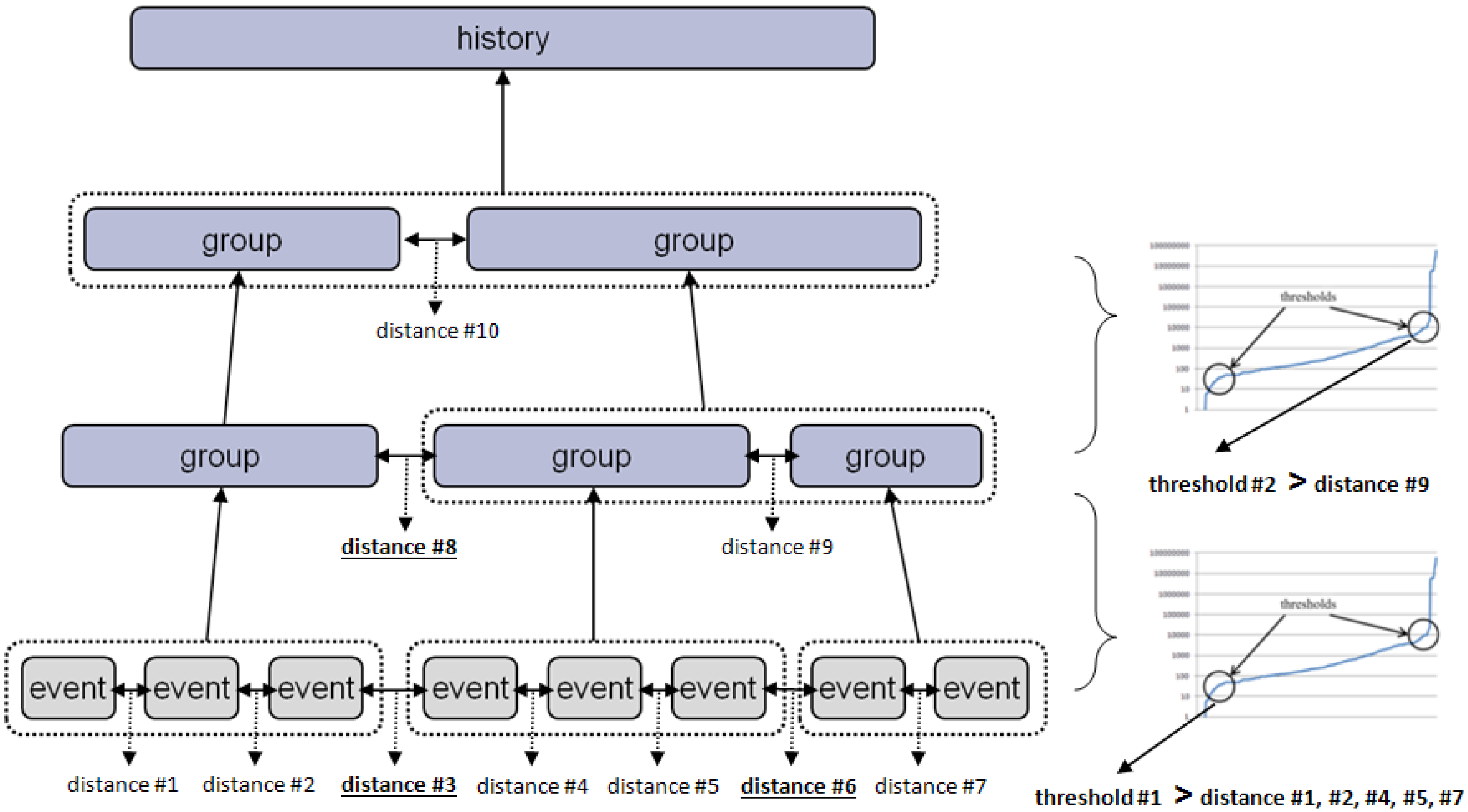


Figure 3. Automatic grouping of history

The distance function can take into account both application-independent (e.g. a timestamp) and application-dependent characteristics of events. By default, the time gap between events is used as the distance function. However, to obtain better grouping results, the distance function can be redefined based on the characteristics of a target application.

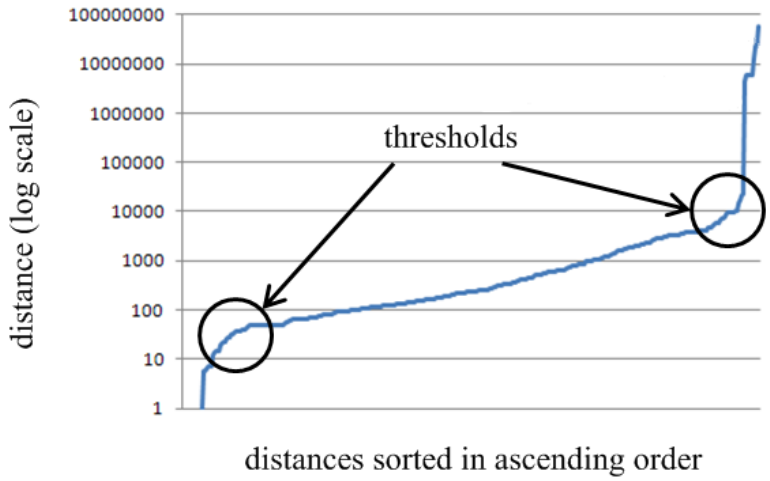


Figure 4. Selecting thresholds for history grouping

The second step in clustering events is to select the distance thresholds used to group events. The more thresholds selected, the deeper the resulting tree of events. CoActIVE dynamically selects the thresholds based on an analysis of the distribution of pairwise distances between consecutive history events. The curve in Figure 4 provides an example of one algorithm for selecting thresholds. The pairwise distances are sorted in ascending order and the dramatically increasing values in the distance curve are

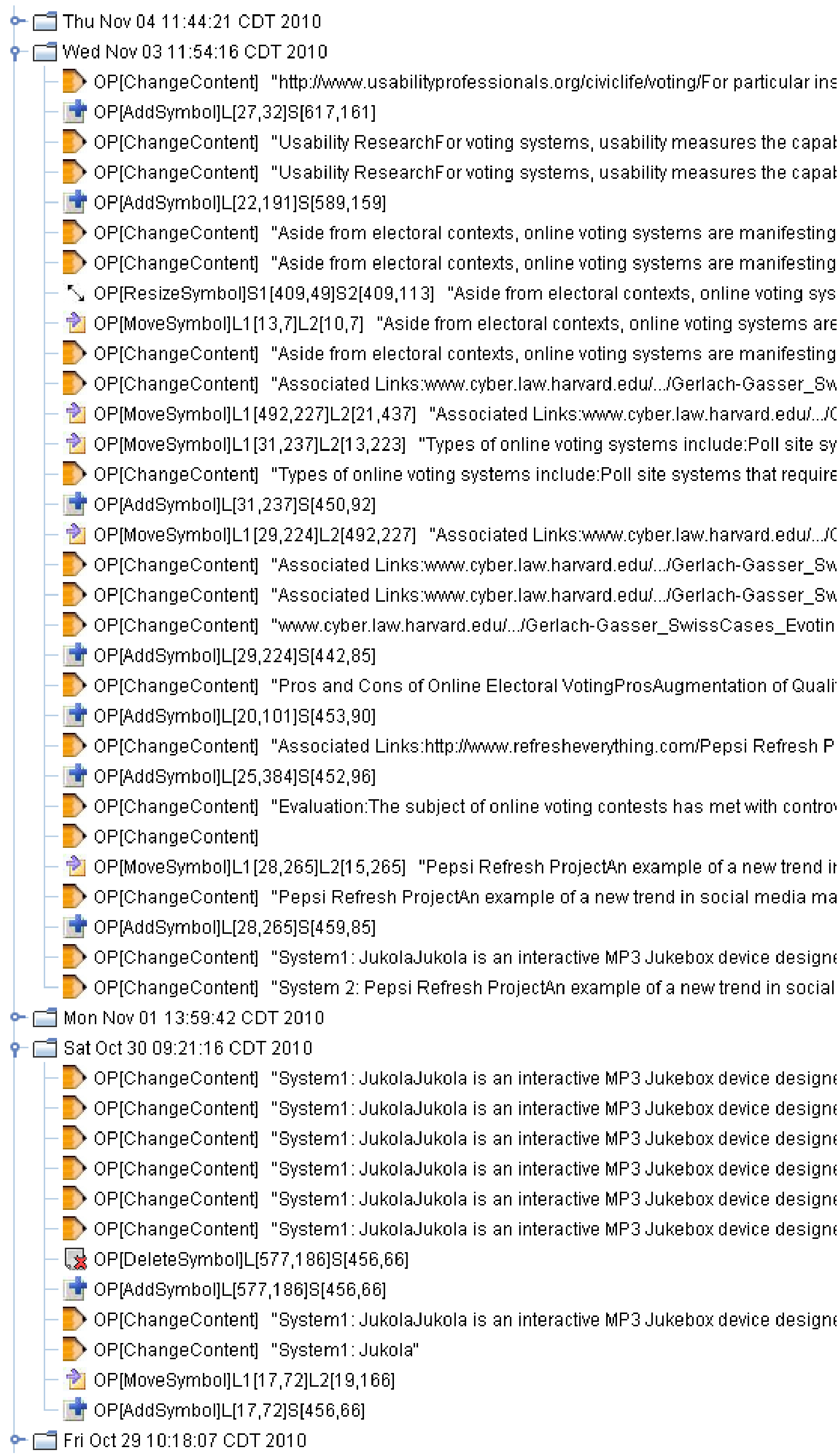


Figure 5. History events at system-level

selected as thresholds. Such an algorithm could be automated by looking at the generated distances and identifying thresholds that match users' desired behavior.

Once thresholds are selected, the consecutive events whose distances are below the thresholds are grouped together. This step is repeated until all thresholds are used for grouping. For example, in case of Figure 4, this step needs to be repeated two times for two thresholds from the lowest value. Since two threshold values are selected, the result would be a three-level hierarchical structure as shown in Figure 3. Finally, all the events are grouped into the final event tree that hopefully corresponds to what users would see as activities and subactivities.

Figure 5 is a brief example that shows how history events are represented in CoActIVE. It presents a two-level tree in which leaf nodes represent low-level events and their grouping as parent nodes. Each leaf node displays system-specific information such as an event type. The tree presents recent events at the top, while earlier ones at the bottom. With the system-level events of this tree, there could be difficulty in comprehension and navigation of history.

Figure 6 is the result of hierarchical clustering applied to the system-level events in Figure 5. A multi-level tree is established that could be used to infer task/sub-task structures in history. In the tree, leaf nodes represent low-level events, while internal nodes exhibit higher-level activities. From the bottom of the tree to the top, the granularity of the decomposition of history decreases so that users can understand how their work has been progressed over time at both detailed-level and overview-level.

However, with this tree, the users still need to open and view clustered activities in order to discover what they are about.

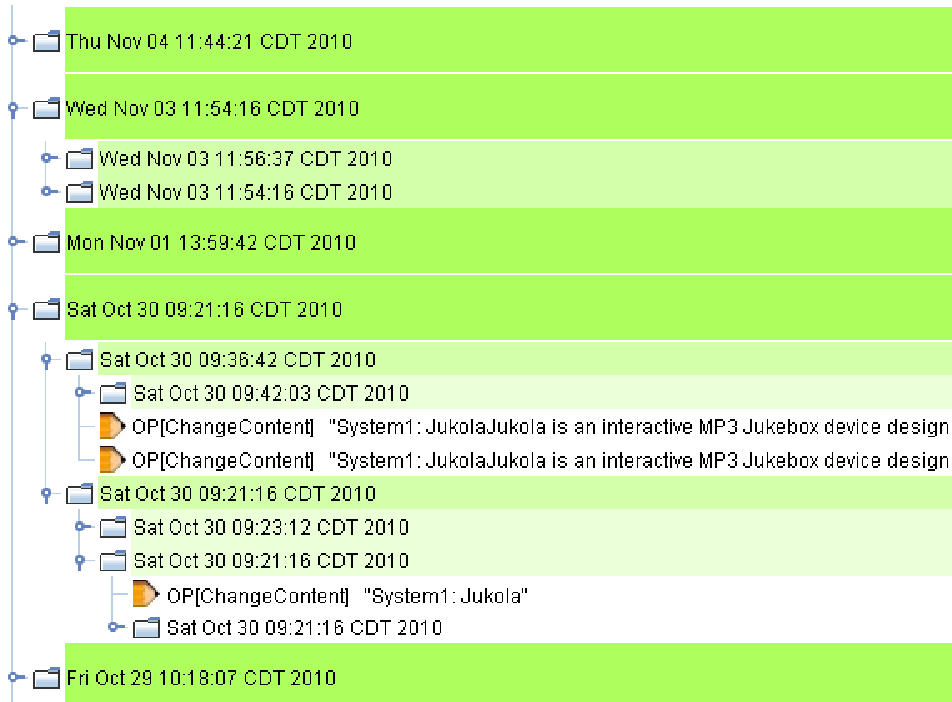


Figure 6. Multi-level aggregation of system-level events

In collaborative work settings, there are a variety of event features that can be helpful in clustering work into meaningful contributions. For instance, in case group members take turns working on the same project, their contribution can be grouped based on a large time difference between their activities. On the other hand, if they work on different parts of the project, the character or location of those parts can be used for grouping. The information about which member executed each event (e.g. username) can be utilized as well.

Each event feature is likely to generate a different cluster tree if used alone as a distance function. Work can be grouped by time, by location, by person, by content, etc. While the CoActIVE software includes a default distance function based solely on time of event, a better result will be achieved by determining an application-specific combination of available features.

5.2.2 Summary Generation and Keyword Extraction

Once a history is clustered, CoActIVE generates a textual summary to represent each cluster. When no other information is available, the time period for the history acts as the summary. CoActIVE includes an optional textual content field for each event that can be populated by an application-dependent function added during integration.

When CoActIVE has content field data, the interpretation module extracts keywords for each cluster by employing the Java WordNet Library (JWNL)[1]. Only nouns are considered after removing stop words from the content. With the selected nouns, their TF-IDF (Term Frequency – Inverse Document Frequency) weights are calculated to suggest significant keywords for groups at each level of a clustering tree. The keywords are then added to the time period to be the summary of an event cluster.

Figure 7 shows a clustering tree identical to that of Figure 5, yet it provides keywords to help users understand the content being manipulated within each grouping. The history used in this example is from the collaborative authoring project in a graduate CSCW (Computer Supported Cooperative Work) class. The project team investigated an

[1] Java WordNet Library, http://jwordnet.sourceforge.net

online voting system, and the grouping tree provides information about what was investigated and how the project has progressed over time.

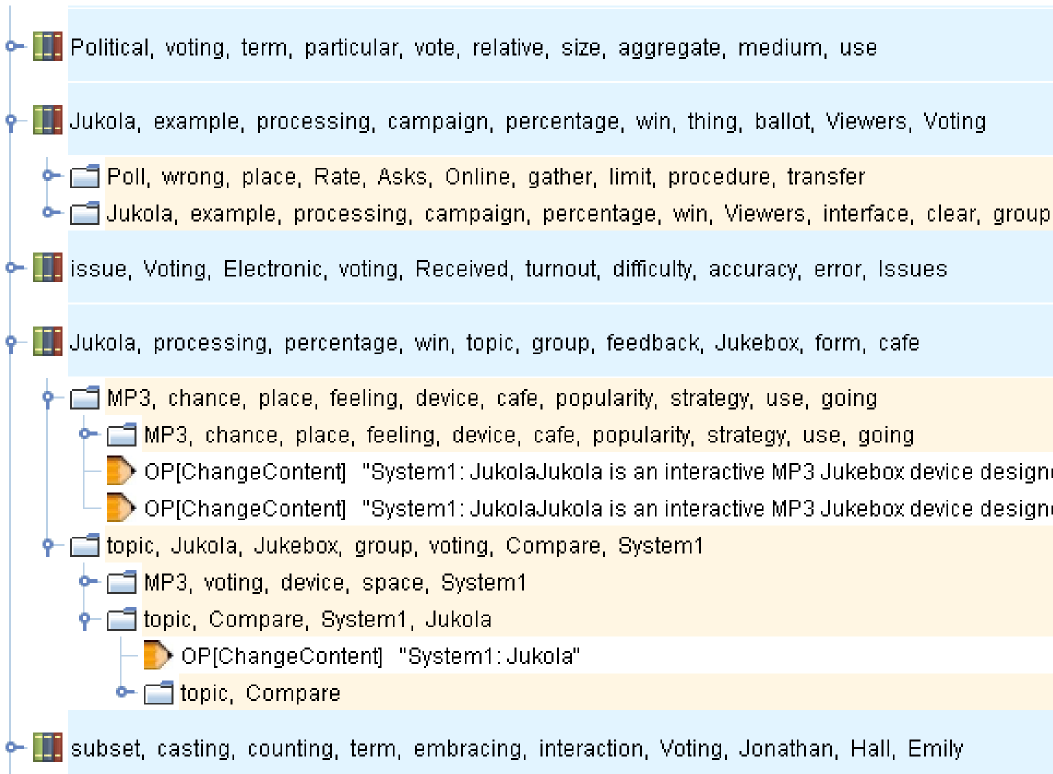


Figure 7. Summary extraction

From the keywords in the tree, a user would expect that there has been some work regarding the issue of accuracy and error in electronic voting, and a voting-based MP3 jukebox named Jukola. Conversely, when the user wants to find the time when Jukola was discussed, the second and the forth nodes start with “Jukola” at the first level would be the candidates to examine. If further investigation is necessary, the user could unfold a node and check the summary of its child nodes. The two opened nodes in

Figure 7 present this situation and their child nodes provide more clues as to the aspects of the Jukola system considered during each clustered activity. As the progression from Figure 5 to Figure 7 shows, clustering and keyword identification may be useful in providing access to long histories.

5.3 Visualization Components

CoActIVE employs multiple visualization techniques to assist users in locating particular activity and understanding the history of collaborative work. When history becomes large, visualizations that represent the whole history in detail are not possible. Instead, visualization components must either select specific points in the history or highly abstract characteristics of the history to present. By providing both high-level and low-level navigation components, users can navigate history via overview and detailed views simultaneously. These views are tightly-coupled to help the users orient themselves during history navigation.

This section includes several screenshots of visualization components from CoActIVE's integration with the Visual Knowledge Builder (VKB). However, the screenshots are the default visualizations of CoActIVE, whose only assumption is that a thumbnail generator for individual states can be provided by the application. Visualizations customized for VKB are presented in the next chapter (Chapter VI).

The visualizations included are the branching history viewer, the history session viewer, the history interpretation viewer, and the filmstrip visualization with associated textual summaries and visual summaries.

5.3.1 Branching History Viewer

The branching history viewer provides an abstract view of branching history by visualizing its logical structure on the left and temporal order of each branch on the right (see Figure 8). It employs a tree structure to represent branching history logically, and the colored-bar is shown for each branch to provide a sense of the order of history branches.

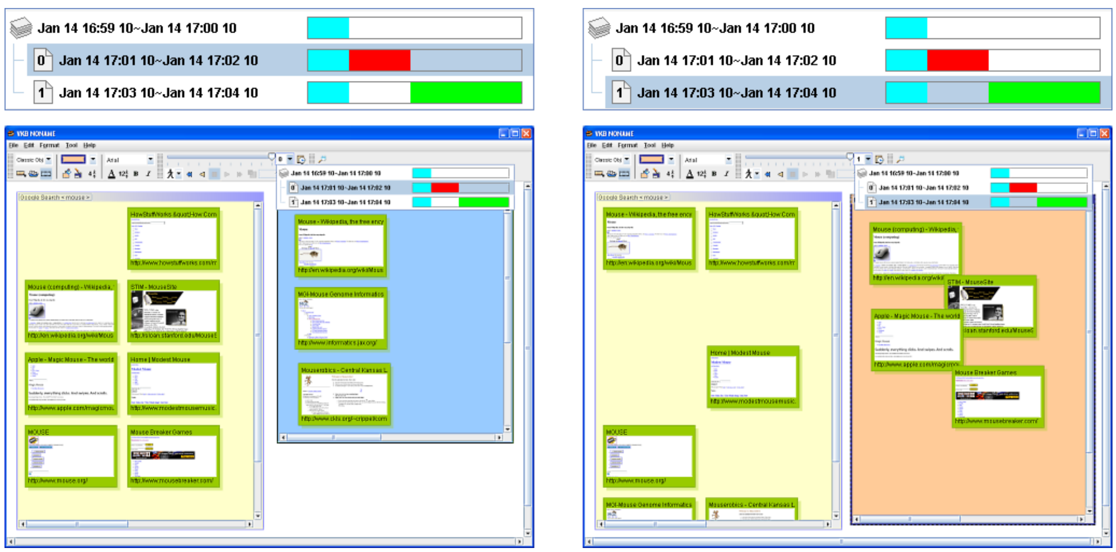


Figure 8. Overview of branching history

In Figure 8, the top-left tree view indicates that currently a user is investigating the first history branch (version 0), and the screenshot below shows a corresponding workspace of VKB. When the user selects the other branch in the view, VKB rebuilds its current workspace to match the workspace with the selected branch. In order to do this, the history manager applies an undo operation repeatedly until its navigation id reaches the latest time where the current and the target branches are separated. Then, a redo operation of the selected branch is repeatedly applied until the navigation id reaches the end of the branch. The right side of Figure 8 reflects the result of switching to the second branch (version 1). Once the current working branch has changed, all visualization components (e.g. the history session viewer, the history interpretation viewer, and the filmstrip visualization) will refresh to reflect this change.

5.3.2 History Session Viewer

The first and most basic, detailed visualization is the history session viewer. It is almost same as the history interface found in VKB 2 [Shipman, Hsieh, Maloor and Moore 2001]. The viewer shows a list of recorded events via a history tree component (see Figure 9). The history tree presents those events via a two-level tree where leaf nodes represent system-level events and their parent node is a grouping of those events based on a predefined time gap. Each tree node displays an event type (e.g. MoveSymbol), operation details (e.g. move from L1[17, 72] to L2[19,166]), and content from which the event was applied. In the tree, recent events are shown at the top, while earlier ones are at the bottom.

5.3.3 History Interpretation Viewer

The history interpretation viewer presents the results of history interpretation module similarly to the tree view of the history session viewer. As shown in Figure 10, top-level nodes of the history tree represent the highest level clusters identified. Users investigate more detailed activity by unfolding the nodes and their descendants. Keywords are provided for each group of events in the tree.

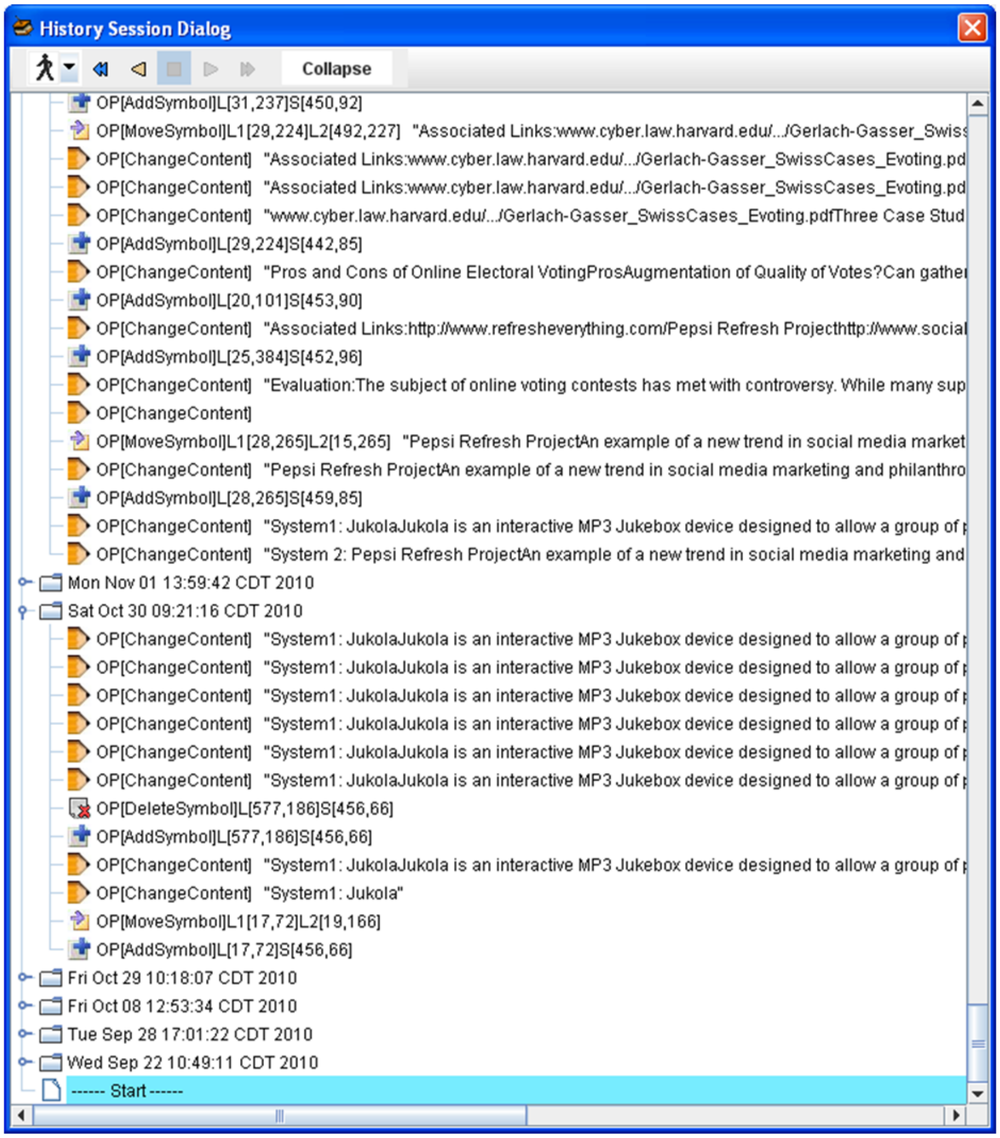


Figure 9. History session viewer

Besides automatic interpretation, the tree allows users to group/ungroup nodes manually to present their interpretation of history. Figure 11 shows that several nodes are selected and about to be grouped via a popup menu. Users can also leave annotations on the group nodes.

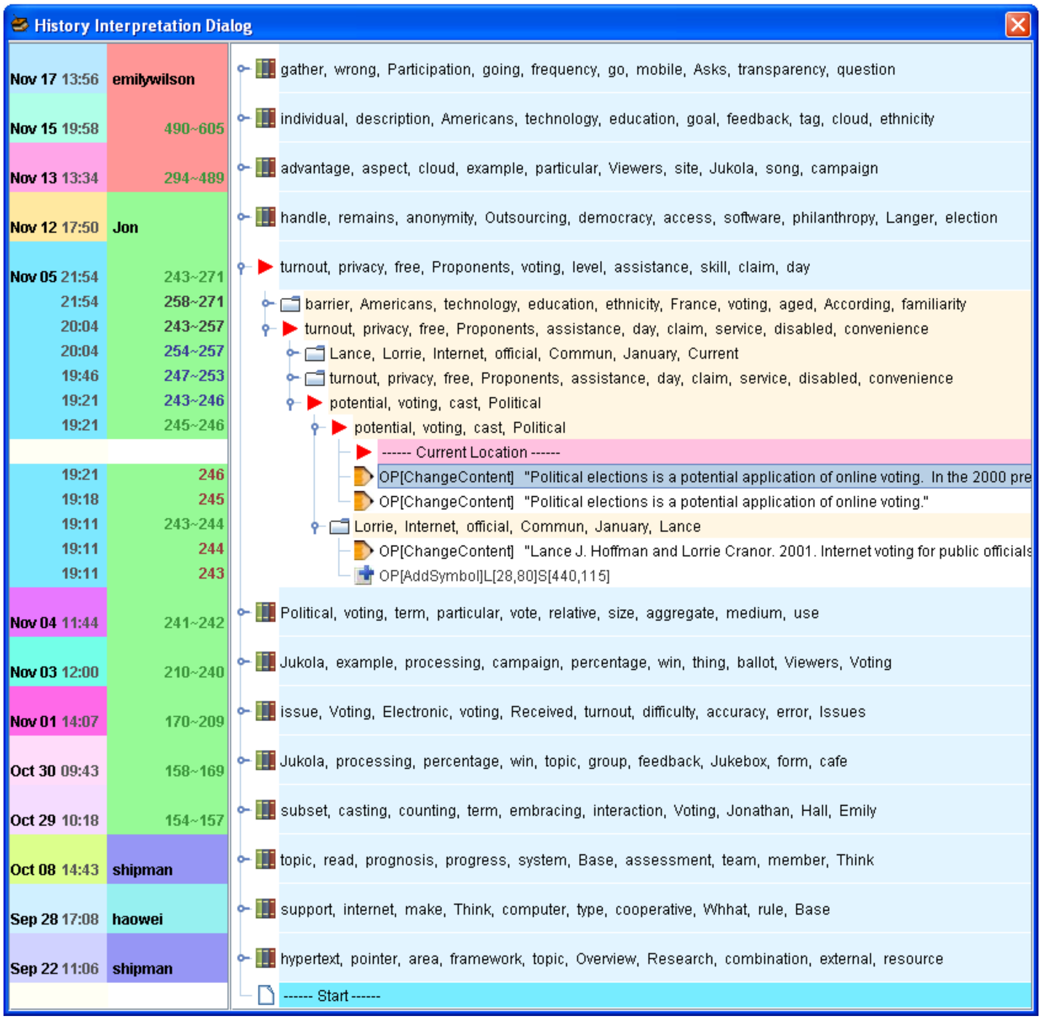


Figure 10. History interpretation viewer

To the left of the tree view is a panel containing information about the time/date of the activity and the user(s) involved in the activity. This panel borrows a chronological table metaphor as often found in history textbooks to amplify user comprehension via color coding and awareness techniques. It employs different colors for each session (left) and each user (right), and it also presents the timestamp (left) and the event id range (right) for the grouping. Figure 10 reveals that the history being displayed includes work by “emilywilson”, “Jon”, “Haowei” and “Shipman” from Sep 22 to Nov 17. “Jon” worked on this work for seven days, after that “emilywilson” took over the work.

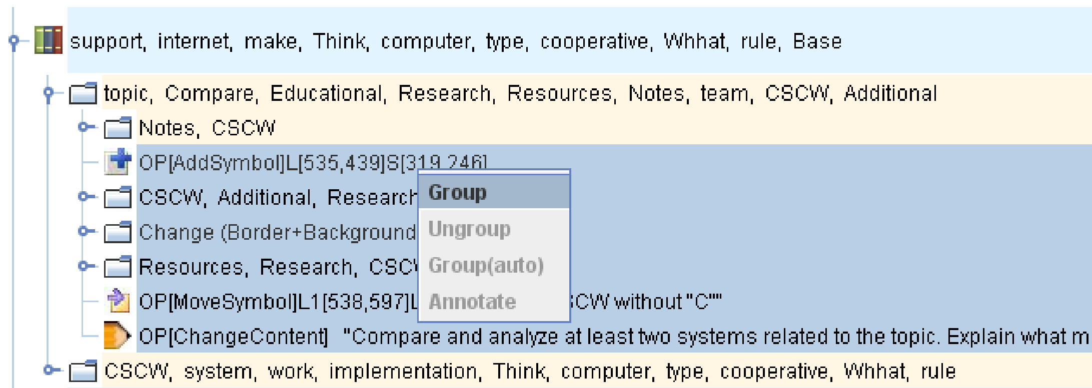


Figure 11. User-defined grouping of history

The panel synchronizes its behavior with the history tree. When a user opens/closes a tree node, the panel refreshes its visualization. Figure 12 shows before and after a node expansion. On the left, the panel shows that the activity on Nov 13 was done by “eboyne” and “Subhadeep” together. When a user expands the node of the activity, the history tree displays two child nodes, and at the same time the corresponding part of the panel is expanded. The right side of the figure presents that “eboyne” worked on the top node and “Subhadeep” worked on the bottom one. Since the child nodes show that two authors worked separately, different color is used in the panel.

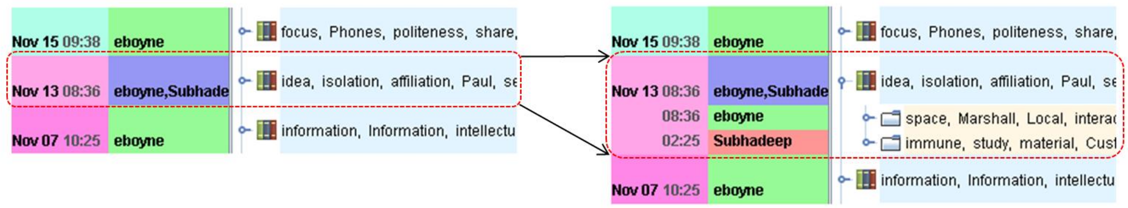


Figure 12. Visualization for an expanded node

5.3.4 Filmstrip Visualization

Filmstrip visualizations of history are a natural way to present change to a visual artifact over time. A thumbnail is a captured image of a target application's state and it is generated every time a substantial event happens in history.

Figure 13 shows the filmstrip with thumbnails of the state of the VKB workspace for each history segment along with associated time, keyword, and user information. On each thumbnail, additional information is displayed, such as a thumbnail's event id (e.g. # 155) and order number (e.g. 5/15), and its time range (e.g. 11/03/2010 08:14 am~11/03/2010 09:33 am). Visual cues are also provided for user location in history. In the figure, the yellow background color on the time range field shows that a user is in the eighth group, and the red vertical line with an event id indicate that the user is located at after the 202nd event was performed.

Additionally, an information panel for each segment is provided under the filmstrip. It is composed of three areas which are for event range, keywords, and author information. The upper area includes the event range of a group and a "details" button for more information on the group. Keywords are acquired from the interpretation

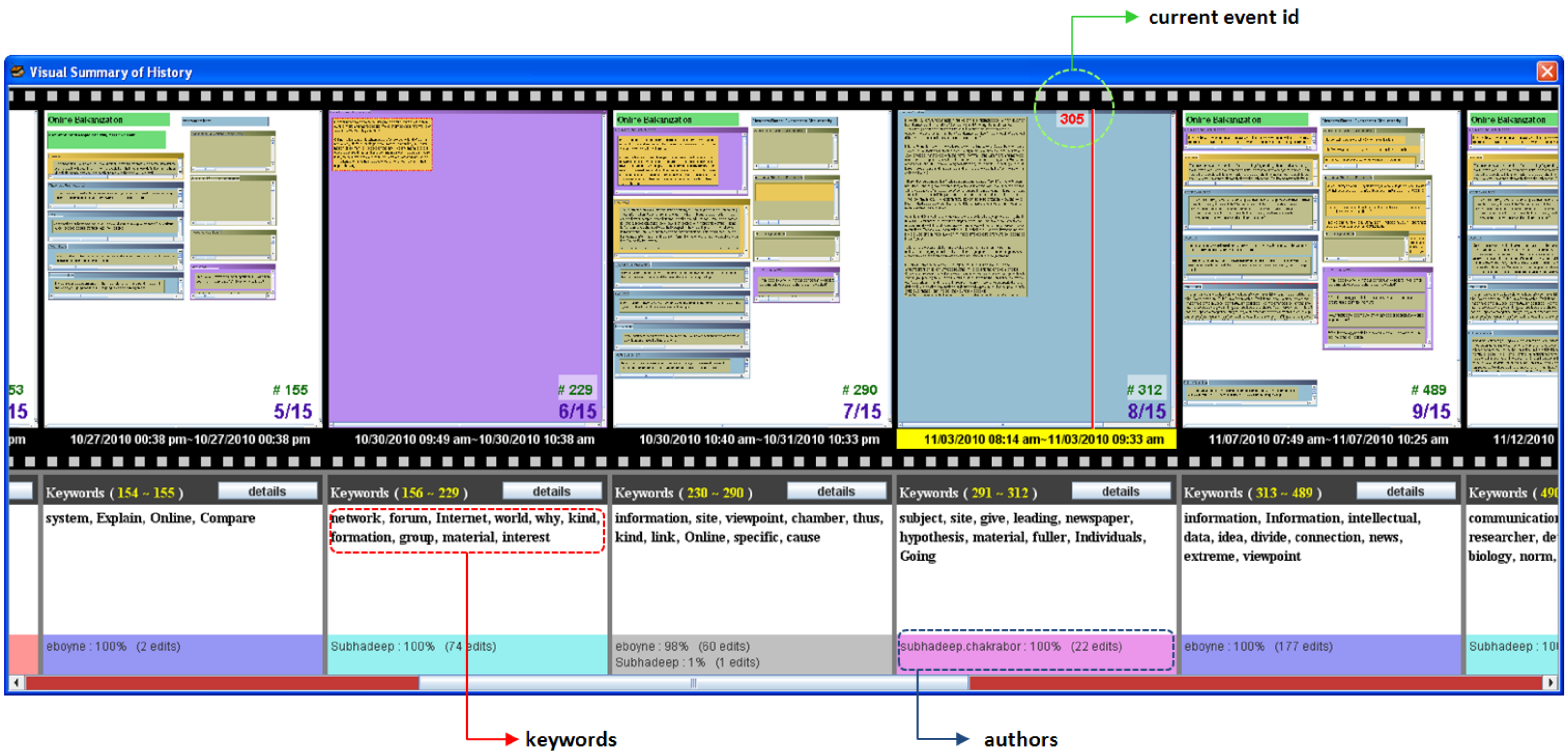


Figure 13. Filmstrip visualization

module and shown in the middle. The bottom area is for author information which includes the username and number of edits for each author active during the time period.

The cluster tree from the history interpretation module is used to decide which images to include in the filmstrip. In order to limit the amount of scrolling required of users, the visualization includes a maximum on the number of segments presented (current default is 16). However, this number can be reassigned or a particular level selection policy could be used instead. Figure 14 shows the process of generating thumbnails by utilizing a cluster tree. In the figure, the circled level has 15 nodes, and it is the highest level that has less than or equal to 16 nodes.

After the level selection, a thumbnail is generated for each node. Because the thumbnail represents a history segment, the filmstrip captures it after executing the last event in the segment. This assumes that the latest status of a segment is the result of performing all events in the segment.

Under each image is a "details" button for accessing more information about a particular history segment via an additional window. In the following study, there were two alternative details views: a textual summary and a visual summary.

5.3.5 Textual Summary

This component presents the textual content of objects and collections modified during the history segment. This provides detail beyond that found in the filmstrip (e.g. thumbnails and keywords) to determine if particular content was involved in a history segment.

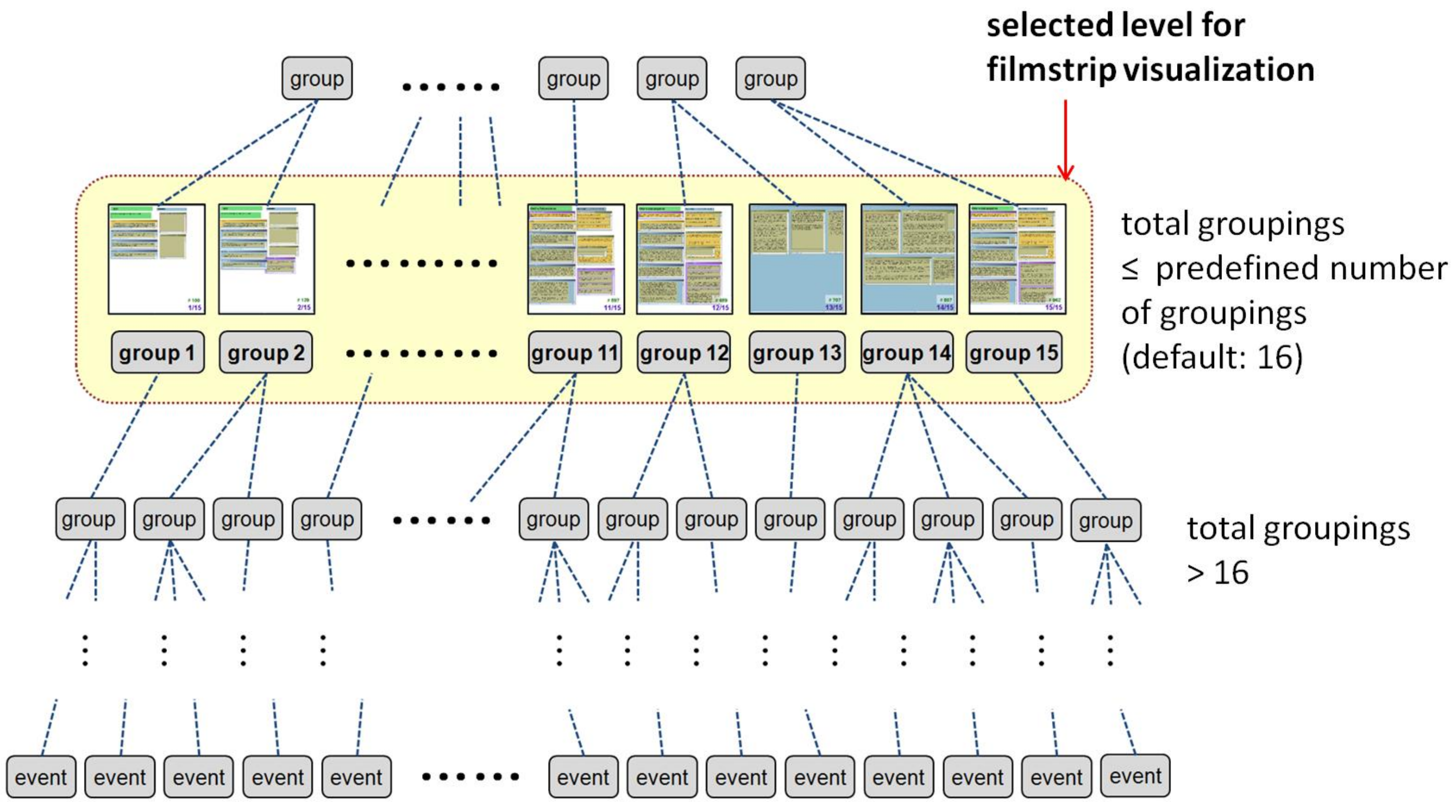


Figure 14. Selection of level of filmstrip visualization

Figure 15 shows an example of a textual summary. It includes all the texts which were worked on in a selected segment. For additional information, a label indicating if a text is created or deleted is provided. The red colored label titled “New” is for newly created content, while the gray colored one titled “Deleted” is for deleted content. The content without the label means that it was created previously and updated in the current working segment.

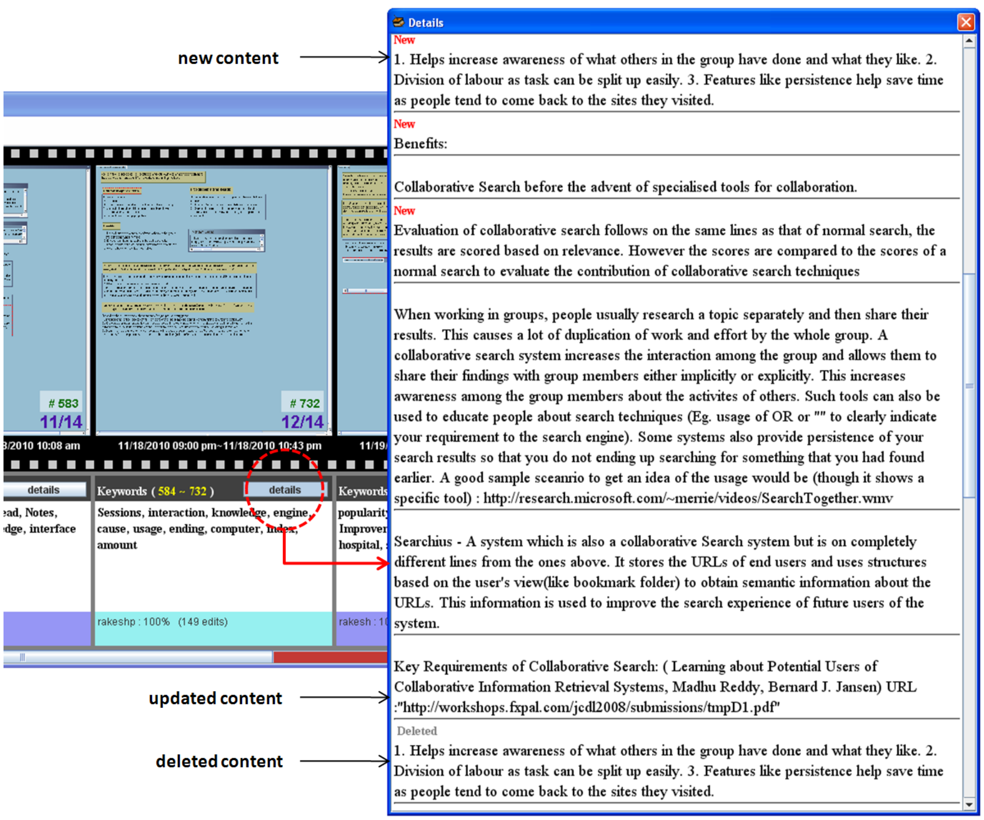


Figure 15. Textual summary

5.3.6 Visual Summary

The filmstrip component provides a visual overview of the users' work process. The visual summary provides more detailed visual cues regarding specific user activities via a series of thumbnails that can be thought of as a more detailed filmstrip.

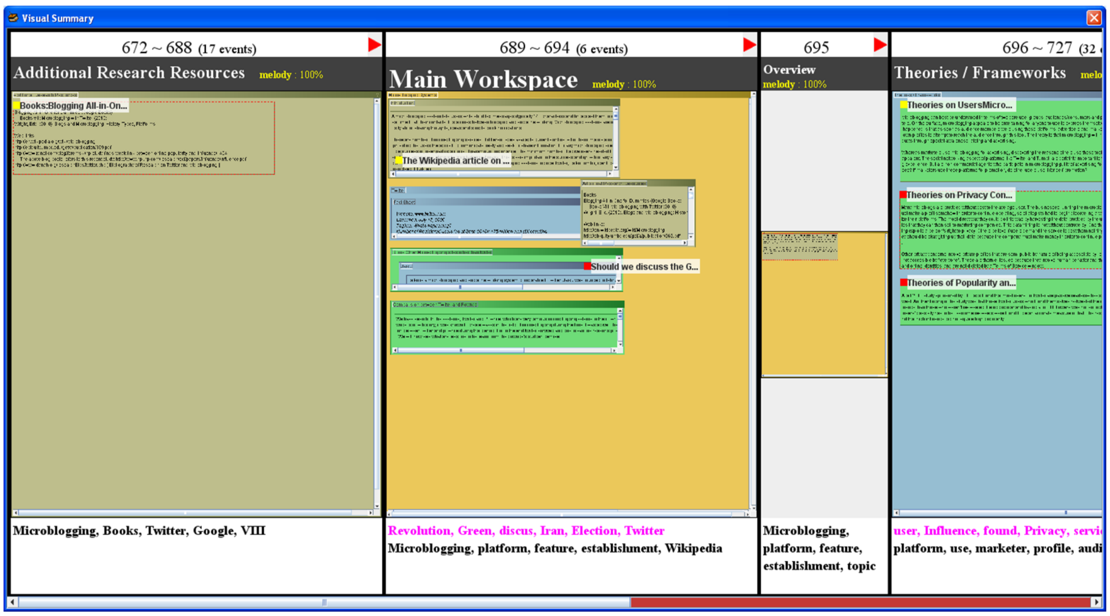


Figure 16. Visual summary

The main advantage of the visual summary is that each user action is guaranteed to be visible in an element in the visual summary. To ensure this there are two rules about when a new thumbnail will be included in the visual summary. The first rule is that when a user action cannot be part of the same thumbnail as the previous user action, a new thumbnail is created. The second rule is that the sequence of edits shown in a single thumbnail will not cross subclusters in the tree. Thus, a new thumbnail will be generated for each subcluster regardless of whether the subcluster could be presented in the same thumbnail as activity from the prior or subsequent subcluster. The overall result is that when a small number of subsequent actions take place in the same region, a single thumbnail is included (e.g. the middle of Figure 16).

Figure 16 shows a visual summary. Similar to the filmstrip visualization, it presents a list of thumbnails with keywords, but each thumbnail includes visual cues indicating user activities during the period specified at the top. For the period with few user activities, the component provides a small sized thumbnail and does not provide visual cues. Since employing visual cues depends on a target application, this component requires customization during CoActIVE's integration with an application.

5.4 History Playback Interfaces

The playback interfaces (see Figure 17) enable users to animate the process of work in a target application. This component provides play buttons for animated history play and a slider for random access of history. At the bottom of the slider, tick marks are used to visualize the quantity of history events. The buttons run like those of VCR, so that users can play and rewind history event-by-event or consecutively. The speed of history play can be specified as well through the play speed combo box. The playback interface also presents the timestamp, navigation id, and username for the event that created the currently displayed state. The navigation controller is similar to that of VKB 2, but it works with the history filters where users can define the scope of history navigation.

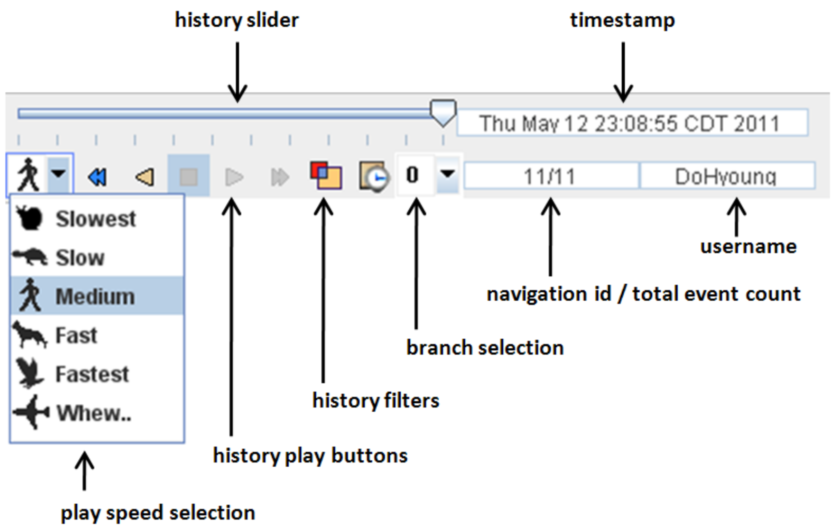


Figure 17. History playback interfaces

The playback interface also provides access to the branching history viewer. In Figure 18.a, the knob is positioned at the end of the history slider, the branch selection combo box displays “0”, and the username is “DoHyoung”. This means that “DoHyoung” was the last user working on branch 0. Then, another user, “Hyoeun” navigates back to a prior state (the navigation id and the knob shows this activity in Figure 18.b), and creates a new branch by performing series of new events. The popup of the branch combo box in Figure 18.c shows the resulting branching structure for the collaborative work. Lastly, another version of the work is created by “Claire” by starting a new branch partway through Hyoeun’s branch (see Figure 18.d).

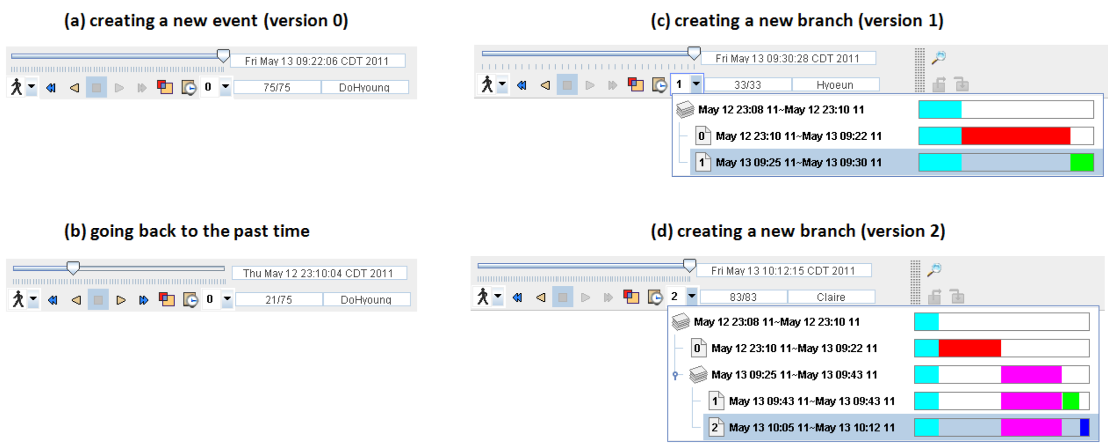


Figure 18. Use of playback interfaces

5.5 History Filters

History filters are used to define which history events are included in the visualizations presented to the user. For example, if a user wants to navigate the history of a certain user's work, the history manager retrieves only that user's work history by masking the work of others and then the visualization components update themselves based on the filtered history. Figure 19 presents the current default filter component that supports application-independent filters such as a username and event type.

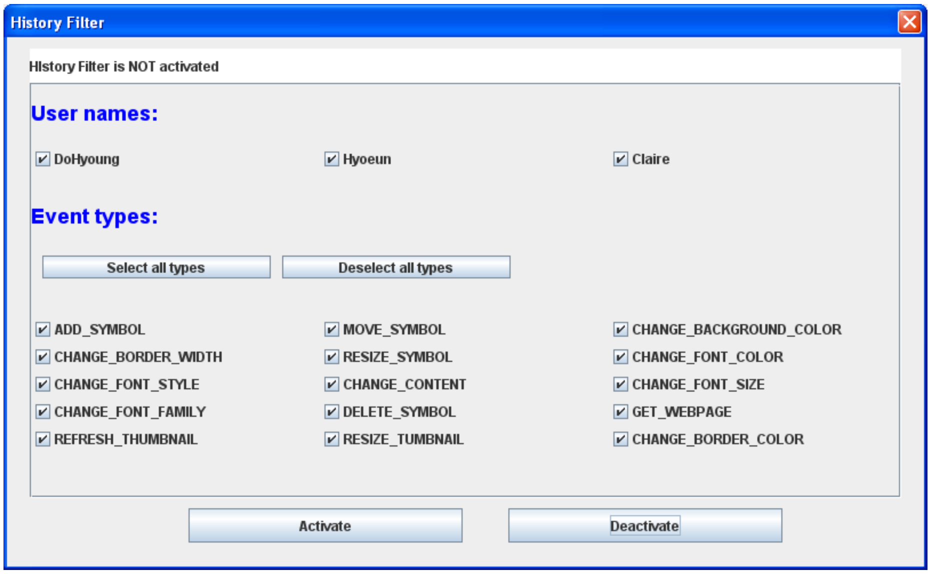


Figure 19. History filters

CHAPTER VI

CoActIVE LIBRARY

CoActIVE is developed as a Java library to represent and manipulate the temporal structure of history. Like a usual Java library, its integration can be done by importing “CoActIVE.jar” in a target application’s project. The CoActIVE library contains five sub packages that correspond to the five parts of CoActIVE’s architecture (see Figure 20). The remainder of this chapter is meant as a programmer’s overview for integrating the CoActIVE library in a Java application.

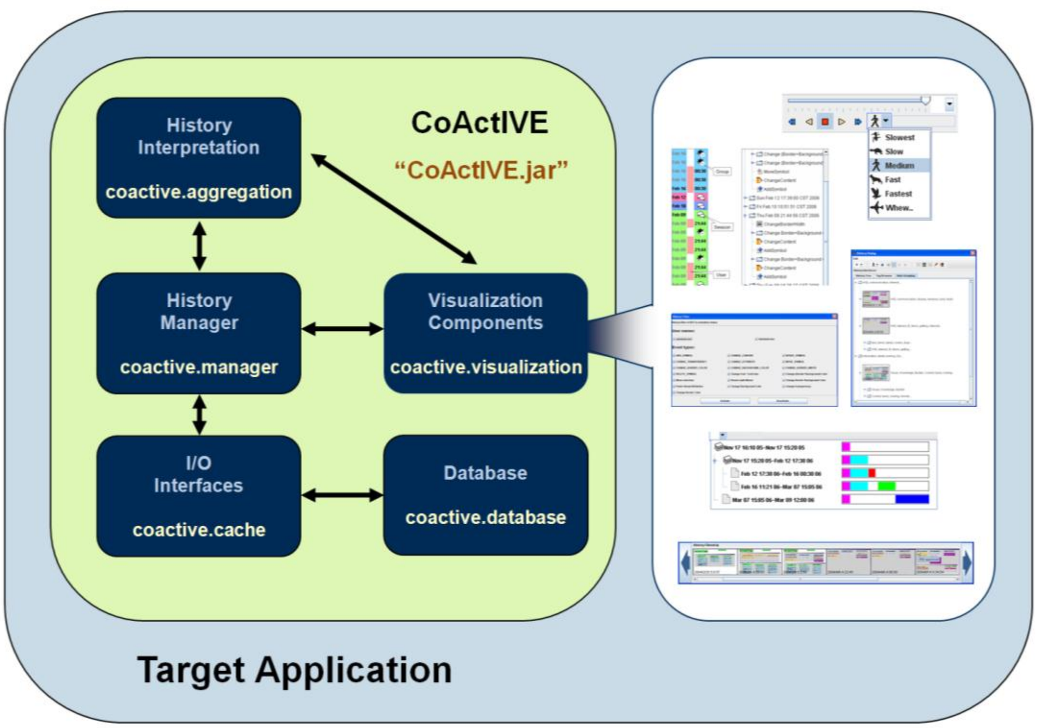


Figure 20. Architecture of CoActIVE library

In the library, the coactive.manager is the main package that controls history information. It reads/writes the history from/to a database through the interfaces in the coactive.cache and the cache uses the coactive.database for database transactions. The coactive.aggregation works with the manager to perform history clustering and summary generation. The coactive.visualization includes a set of history navigation interfaces employing diverse visualization techniques. The interfaces not only present the history information and its interpretation, but also support user interactions with CoActIVE. This chapter describes the implementation and integration details of those packages.

6.1 History Manager: coactive.manager

To employ CoActIVE, a target application needs to invoke the history manager (HIMEHistoryManager class or its extension) in the coactive.manager package when it is initiated. However, prior to integration, the HIMEEventStruct class or its extension needs to be defined because all classes in CoActIVE employ it to handle a history event. The event is the basic unit that represents user activity happened in the application and is called “a system-level event” in the previous chapter.

6.1.1 History Event

When defining a history event, a developer is responsible for deciding what event types and information are necessary for undo and redo operations in an application. This task is important since CoActIVE cannot construct the exact state of previous and current work without properly designed events. Also, some care is necessary in deciding what counts as an event. For example, if an application saved every mouse movement event provided by the operating system during editing, the size of the history would increase unmanageably.

Figure 21. UML diagram of HIMEEventStruct class

Once the design of the application's event is decided, it can be implemented via the HIMEEventStruct class. Figure 21 shows the UML diagrams of the class. It is an abstract class which includes application-independent attributes such as id, event type, task, username, and timestamp. The id is assigned in ascending order starting with the FIRST_UID (1001) when new event occurs regardless of a branching action in history. The class includes two abstract methods – undo( ) and redo( ) – which need to be defined to call the application-specific undo and redo operations.

If the event needs customization based on additional information, a developer has to create a new event class as an extension of the HIMEEventStruct class. CoActIVE can handle the customized event since the event is represented as a generic type variable. When user activity occurs in an application, it has to be represented as a single or a series of events and passed to the history manager via the addHistoryEvent methods. Then, the history manager fires an internal event (HIMEEvent class) to inform relevant components in CoActIVE.

6.1.2 History Navigation

The history manager maintains events in an event list and utilizes them for history navigation. It includes the undo( ) and redo( ) methods that allow users to navigate history in step-by-step manner. To perform undo and redo operations, those methods call the undo( ) and redo( ) methods in an event (HIMEEventStruct class or its extension) and fire a navigation event (HIMENavigationEvent class). For those who want to navigate to a certain time of history, the manager also provides undo(target_event_id) and redo(target_event_id) methods by repeatedly applying its undo( ) or redo( ) methods. A developer can use those methods to provide application-specific history navigation making use of CoActIVE's record of events.

To maintain user's location during history navigation, the history manager employs a "navigation id" to indicate the id of the event that was just performed to reach the user's location in history. The event id is also used to point out the target location of undo and redo operations. After an undo or redo operation is performed, the manager fires an internal event (HIMENavigationEvent class), and the navigation id is reassigned to indicate the target location.

The CoActIVE library also includes methods for navigating branching history. For this purpose, the history manager provides two methods. The navigateHistoryEvent(target_version_id) method allows users to navigate to the latest state of a target branch, while the navigateHistoryEvent(target_version_id, target_event_id) is a specified version in which users can specify the time spot they want to visit in a target branch. To maintain a record of the version where a user is located, the history manager employs "version id" that is updated when those methods are performed.

To determine the branching action, the history manager monitors the status of a navigation id. When a new event occurs, the navigation id is examined if it matches the id of a previously happened event. When the id indicates a new version is being created, the manager fires an internal event (HIMEBranchEvent class) and updates the version id. The navigation id is also updated to the id of the new event.

6.2 History Interpretation Module: coactive.aggregation

The ClusteringManager class is the main part of this module that constructs a cluster tree based on an event list in the history manager. By default the cluster tree is created based on the time difference between events, and this is defined in the TimeBasedClusteringModule class. However, a developer can customize the distance function by implementing the ClusteringModule or DistanceBasedClusteringModule interface. To register the customized function, the developer needs to use the registerClusteringModule method, and the registration should be done when the history manger is initiated. In addition, the number of clusters of the clustering tree can be controlled through the MAX_NUM_CHILREN (default is 16) in the ClusteringManager class.

When an application wants to support textual summaries, the KeywordsExtractor class needs to be registered via the registerKeywordsExtractor method in the ClusteringManager class. The KeywordsExtractor class is an abstract class in which the getSummary(from_event_id, to_event_id) method need to be defined since a developer has to specify the source of extracting keywords. If the selected period of history does not contain textual information or the KeywordExtractor class is not registered, CoActIVE presents the time information of that period as the summary.

6.3 Visualization Components: coactive.visualization

The visualization components in CoActIVE can be integrated without any modification or with application-specific adjustments. In the case of no modification, their visualizations only present common attributes such as event types and temporal information. The interface components for the visualizations are designed by employing event-driven techniques so that they can be integrated in a target application independent from other components. Table II presents the classes for visualization components. CoActIVE provides ten basic components and their combinations.

The visualization components can be modified to present application-specific features (e.g. visual summary) as well as creating new visualizations or information types from scratch. When working with the visualization component, a developer needs to define its behavior for internal events in CoActIVE such as event creation and user

Table II. Visualization component classes

Composite Components	Basic Components
HistorySessionDialog, HistoryInterpretationDialog, FilmstripDialog, ControlToolbar	HistoryTree, InfoTreePanel, VersionTree, HistoryThumbnailPanel, ControlButtonPanel, TextualSummary, VisualSummary, NavigationSliderPanel, EventInfoPanel, VersionTreeComboBox

navigation. For this purpose, CoActIve provides the HistoryComponentBase and HistoryComponentBasePanel classes that include a set of abstract methods to guide the developers in creating a proper component. The HistoryComponentBaseListener interface is also provided to configure the component in more detail.

In addition, to improve the legibility of tree visualization (e.g. History Tree), each type of a history event can be visualized by applying a proper icon to a corresponding node. In order to do this, CoActIVE provides a configuration file under the "HIME" folder that is a subfolder of where CoActIVE library is located. The file needs to be configured with a pair of an event type and the name of icon file separated by colon for each line (e.g. ADD_SYMBOL:add.gif), and the "HIME" folder should include the defined icon files (e.g. add.gif).

6.4 I/O Interfaces: coactive.cache

The I/O interfaces are composed of an event cache and a segment cache. They act as transparent bridges between the history manager and the database. In case a developer wishes to extend the history manager (HIMEHistoryManager class), the event cache provides a set of methods for accessing history information in the database (see Figure 22). Through the methods, the developer does not need to care about the implementation of recording history. Like the manager class, the event cache is a generic class to handle not only the default event structure (HIMEEventStruct class), but also application-specific extensions.

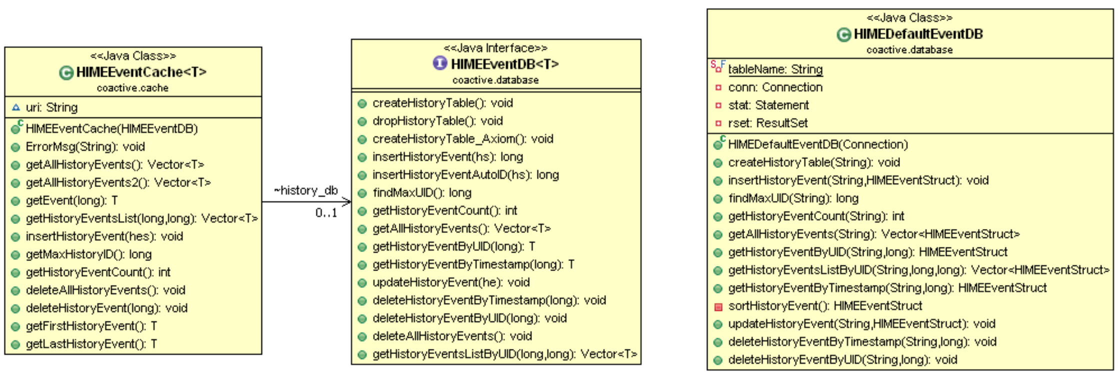


Figure 22. UML diagram of event cache and database classes

The segment cache is used to manage branching history. However, additional work is not necessary for the cache and its segment structure (HIMESegmentStruct class) because the history manager handles the branching structure internally. The branching information is not depending on a particular application.

6.5 Database: coactive.database

For an application that does not have an existing database for recording user history, CoActIVE provides a database module. The module utilizes the Java DB to store the history and the HIMEDefaultEventDB class handles database operations. This class implements the HIMEEventDB interface to provide database access to the event cache (see Figure 22). This interface needs to be implemented when a developer wants to create a customized database. CoActIVE also includes a database module for a segment cache, but it is for internal use to manage branching history.

Figure 23 is a brief UML diagram of CoActIVE. Though the diagram does not describe every part of the mechanism, it shows the relationship among the history manager, the interpretation module, and visualization components. Figure 20 in this chapter is a simplified version of Figure 23.

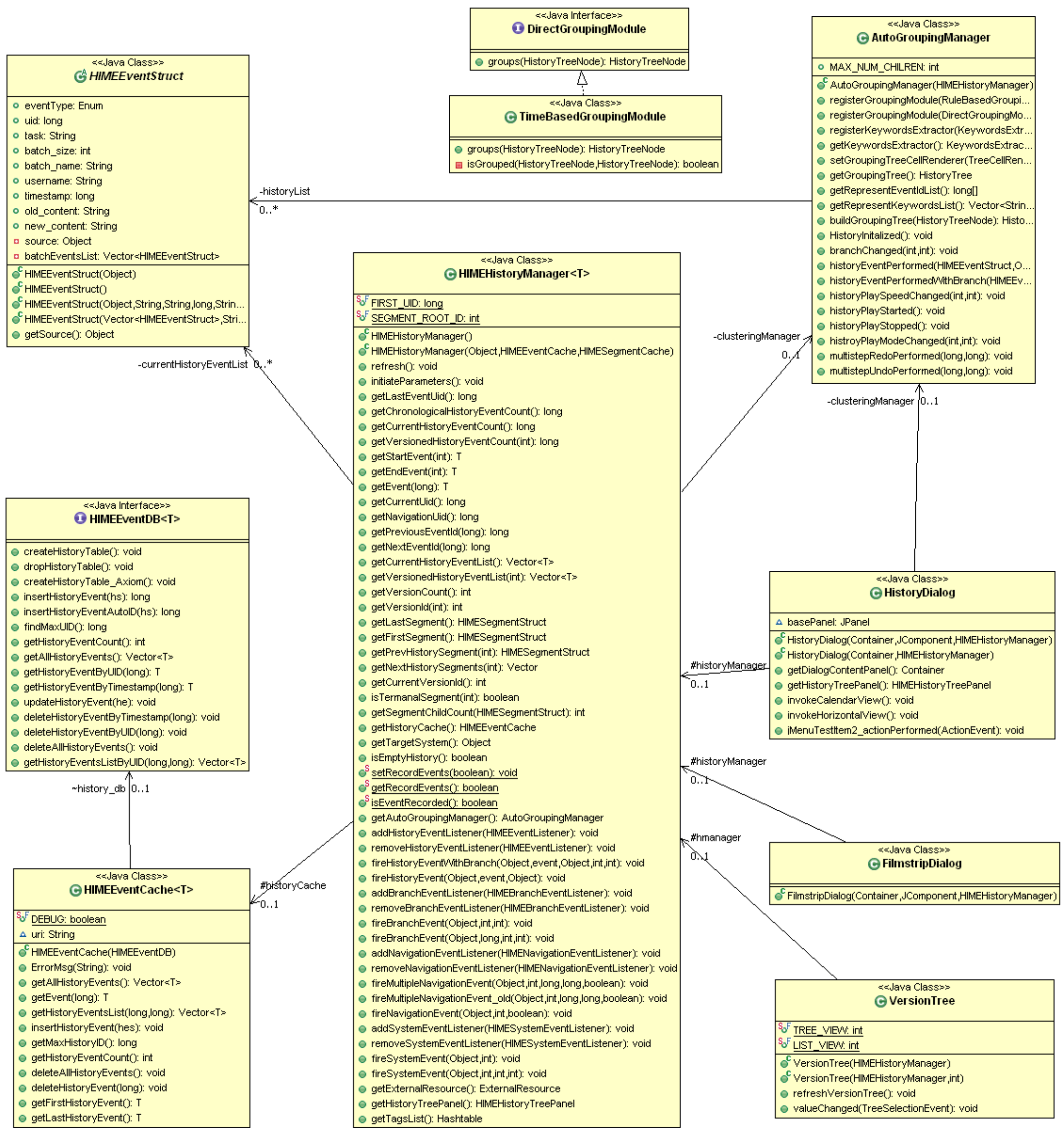


Figure 23. Brief UML diagram of CoActIVE library

CHAPTER VII

INTEGRATION

Currently, CoActIVE is integrated in three applications: the Visual Knowledge Builder (VKB) [Shipman, Hsieh, Maloor and Moore 2001], VKB Server (a Web Service that supports collaborative work on VKB documents), and the Design Exploration (DE) software design tools [Moore 2007]. Among those applications, this chapter mainly describes the integration of CoActIVE in VKB where user evaluation is performed.

7.1 The Visual Knowledge Builder

The Visual Knowledge Builder (VKB) is a spatial hypertext system developed to support collaborative knowledge building. The system provides a two-dimensional workspace where users author and collect information, and share this workspace with their collaborators. In the workspace, information is represented as information objects which can contain text, attributes and values, links to files or URLs, and images. Users can assign visual characteristics (e.g. border/background color and border width) to objects and place objects in a hierarchy of two-dimensional spaces called collections. Through the placement of the objects with visual attributes, users can categorize the objects and interpret collaborators' representation of information. VKB was chosen as the first application to be modified to use CoActIVE because it already contained a history mechanism and interface that could be compared to CoActIVE. Figure 24 shows the spatial hypertext representation of a personal webpage that was worked in VKB 2 (current release).

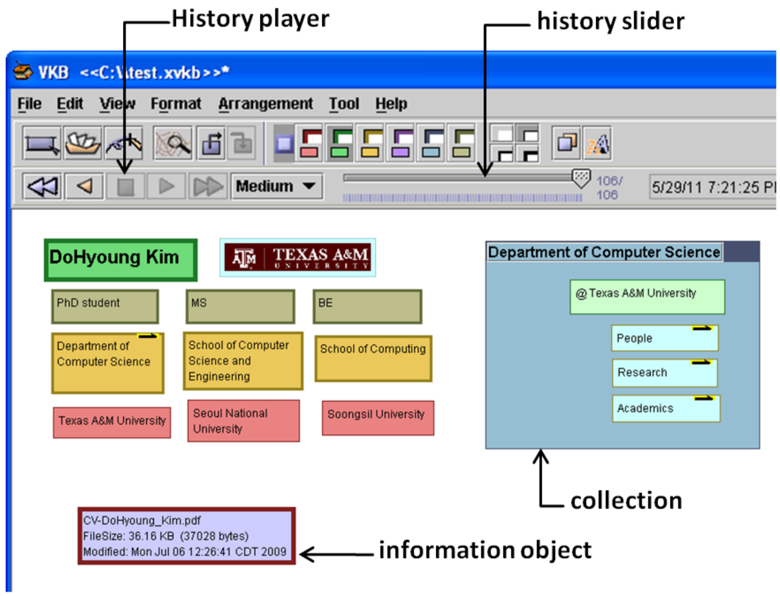


Figure 24. Visual Knowledge Builder 2 (current release)

VKB includes a navigable history mechanism that provides four interfaces for navigating back and forth through the timeline of the workspace. First, the history player (see VCR-like control buttons in Figure 24) allows users to play forward and backward of history in a step-by-step or continuous play manner. The second interface is the history slider (on the right of the history player) that provides random access through the scrubbing of history by dragging its knob. The history session dialog groups the low-level edit events based on a predefined time gap to provide a sense of authoring and editing session (see Figure 25). Lastly, users can go back to a specific event on a particular information object through popup menus in the workspace. These history interfaces allow users to rewind and replay editing actions to recognize the patterns of activity and to examine specific actions in order to better understand content in the information space [Rosenberg 1996; Shipman and Hsieh 2000].

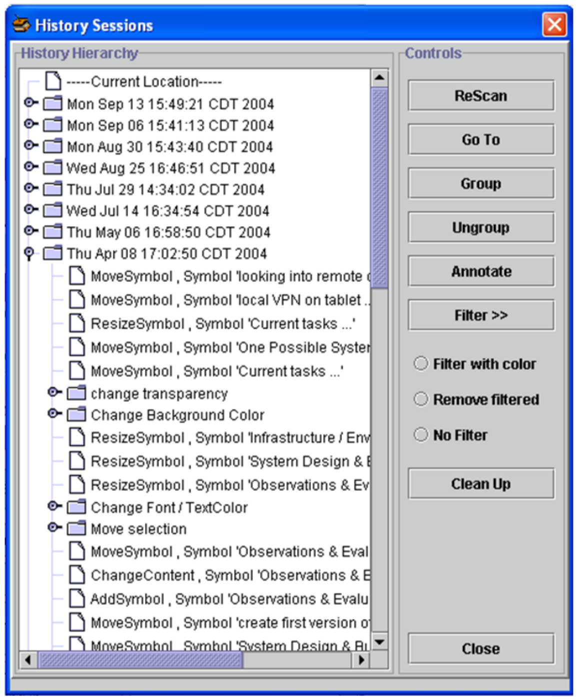


Figure 25. History session dialog of VKB 2

Regarding the history mechanism in VKB, users are left to interpret system-level records to understand previous work in workspace. Moreover, as history records increase, navigation to the moments of particular activity becomes more difficult. Also, editing the previous state of a workspace causes all future events to be removed, losing a potentially valuable source of insight into the thought process that was undertaken. For these issues, CoActIVE provides diverse visualizations based on the interpretation of system-level history. The next section describes the integration of CoActIVE in VKB to improve history support in collaborative work.

7.2 Integration in Visual Knowledge Builder

The CoActIVE library was integrated with VKB through the programming interface described in Chapter VI. CoActIVE's abstract event class (HIMEEventStruct) was extended to attach specific information for VKB event types, and the undo/redo operations of each event were defined. The distance function was established by considering VKB-specific features for better interpretation and the history visualization components were customized based on the visual nature of the medium. In order to manage the customized events, the database interface (HIMEEventDB) was extended.

7.2.1 History Event Handling

In addition to the required information found in CoActIVE's event data structure, VKB's preexisting event representation includes the content, visual properties, and locations of objects being manipulated. The content of the objects and collections affected by an event is utilized to generate a textual summary of activity and to locate keywords to display with events. There are 24 event types for VKB, 15 for the current release (VKB 2) and 9 new events included in the next release (VKB 3) (see Table III).

7.2.2 Distance Function for Clustering

In order to achieve a better clustering of low-level events into higher level activities, VKB redefines a distance function by considering additional event features and visual-spatial characteristics. Currently, two distance functions are provided. One is a rule-based approach, and the other is a *spatio-temporal* approach.

The first approach employs human-defined rules based on the observations of how users interact with VKB. It assumes that consecutive events for a particular object

Table III. Event types of VKB

VKB 2 / VKB 3	VKB 3
ADD_SYMBOL, DELETE_SYMBOL, MOVE_SYMBOL, RESIZE_SYMBOL, CHANGE_BACKGROUND_COLOR, CHANGE_BORDER_COLOR, CHANGE_BORDER_WIDTH, CHANGE_FONT, CHANGE_FONT_COLOR, CHANGE_FONT_FAMILY, CHANGE_FONT_SIZE, CHANGE_FONT_STYLE, CHANGE_CONTENT, CHANGE_TRANSPARENCY, CHANGE_ATTRIBUTE	GET_WEBPAGE, RESIZE_THUMBNAIL, ARRANGE_TITLE, ARRANGE_THUMBNAIL, ARRANGE_URL, APPLY_STYLE, CHANGE_TITLE, CHANGE_ANNOTATION, CHANGE_URL

are closely-related. Therefore, consecutive events on a particular object are assigned a distance less than the lowest level threshold so they will be grouped together at the lowest level of the resulting tree. Similarly, this distance function groups all consecutive events that occur within a particular collection at the next higher level of the event tree. The rule-based approach shows how symbolic logic can be used in the distance function. This approach could be improved by combining such results with the analysis of temporal, spatial, or conceptual features.

The *spatio-temporal* approach defines the distance as a weighted sum of a spatial distance and a temporal distance between consecutive events. The temporal distance is simply a time span between events, while the spatial distance is complicated due to the hierarchy of collections embedded in a VKB workspace.

$$\textit{distance} = \alpha \times \textit{spatial distance} + \beta \times \textit{temporal distance}$$

$$\alpha, \beta : \textit{coefficients}$$

The spatial distance between events is computed to be the minimum distance between the two areas of effect of the history events. The area of effect (AoE) of an event is defined as the bounding rectangle that includes all the affected objects before and after the operation. Thus, the AoE for a color-change event on single object is the area of that object. The AoE for a move object event is the smallest bounding rectangle that includes the initial and final position of the object. When an AoE spans multiple collections then the event includes an area of effect in both collections and the spatial distance is the minimum of the distances calculated to each of these areas of effect. To compute the minimum distance, the Euclidian distances between each four points of a previous event's AoE and each four points of a current event's AoE are calculated. The spatial distance is the minimum among these distances. Figure 26 shows that each spatial distance of areas A and B is different based on their co-location. In the case that two consecutive AoEs are overlapped, the minimum distance becomes 0.

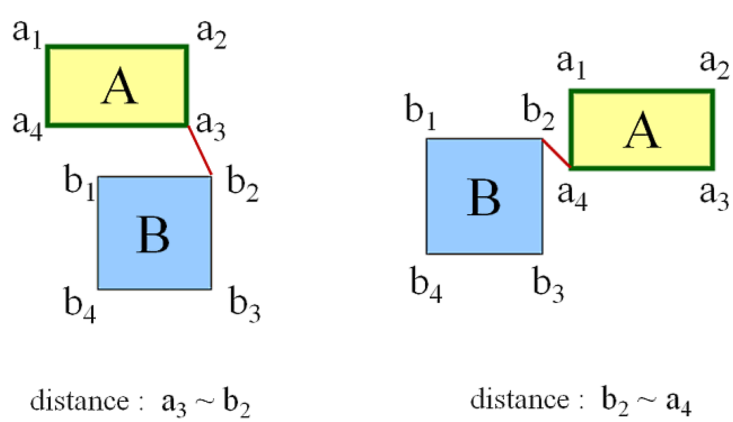


Figure 26. Calculating spatial distance

Collections in VKB form a hierarchy of spaces and subsequent events sometimes do not include AoEs in the same collection. In such cases, the spatial distance is computed by counting the number of collection traversals that must take place to navigate from one AoE's collection to another. This is the distance between collections in the tree representing the collection hierarchy. The spatial distance of the AoEs is then a constant multiplied by this number of traversals plus the spatial distance between the highest non-shared collections in the lowest common parent collection.

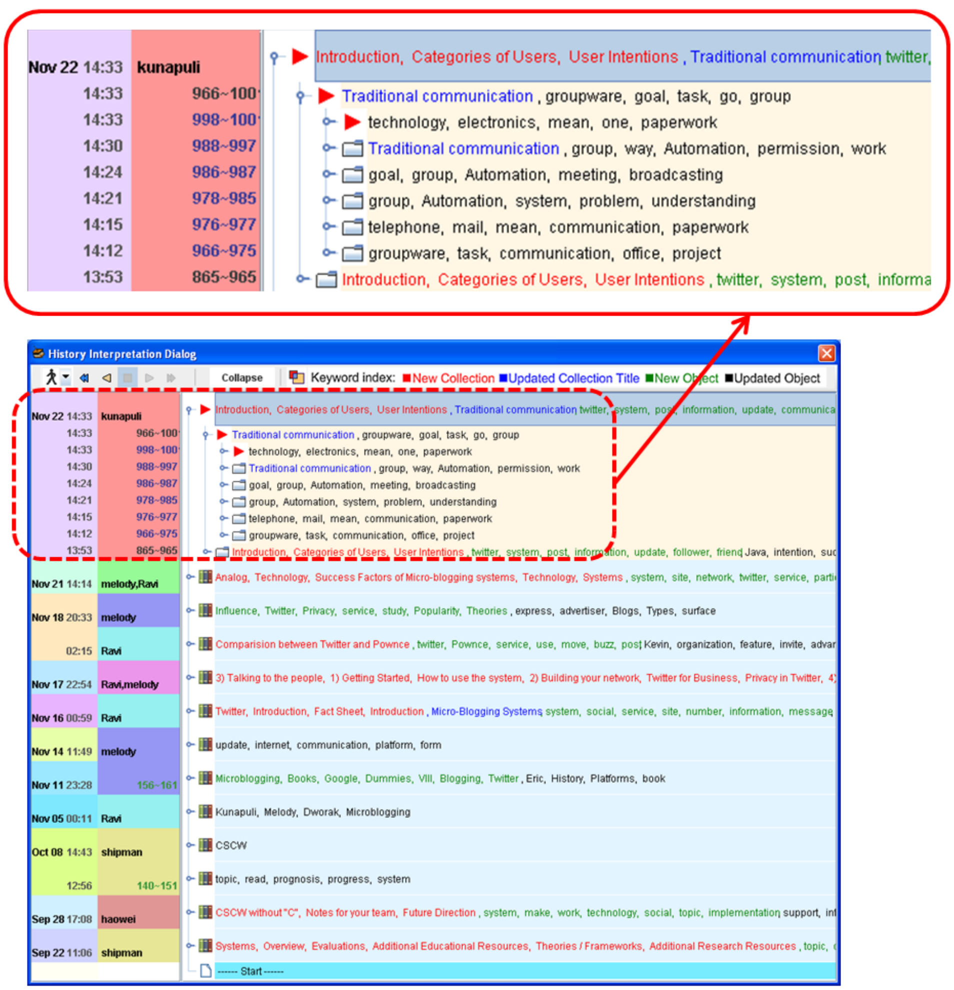


Figure 27. Visual cues in history interpretation dialog

7.2.3 Visualization Components

Among visualization components in CoActIVE, the history interpretation viewer, the filmstrip visualization, and the visual summary were customized during VKB integration to include color coding for keywords and thumbnails to indicate features of the events or objects being manipulated.

Figure 27 shows their use in the history interpretation view. Color coding provides cues for the role of keywords in the activity represented by the cluster. Red indicates the term comes from the title of a collection created during the activity. Blue indicates a term from a collection title that was modified. Similarly, green and black indicate terms from information objects that are either created or modified. Each component including color-coded keywords includes a legend indicating the meaning of the colors at the top-right (see Figure 27 and Figure 28).

VKB's filmstrip visualization also uses the same visual cues for its summary (see Figure 28). Furthermore, each segment of the filmstrip was tailored to provide a list of collections' titles that were involved in user activity. When a user places a mouse pointer over the thumbnail of the selected group, a small window pops up on it and shows a list of collection titles. The window employs a blue-colored "Updated" label to specify an updated title and a red-colored "New" label for a new one (see Figure 28.b). The title without a label means that particular user activities were applied on the collection of the title (see Figure 28.a). Through the information of the window, the user can determine in which segment a certain collection was involved.

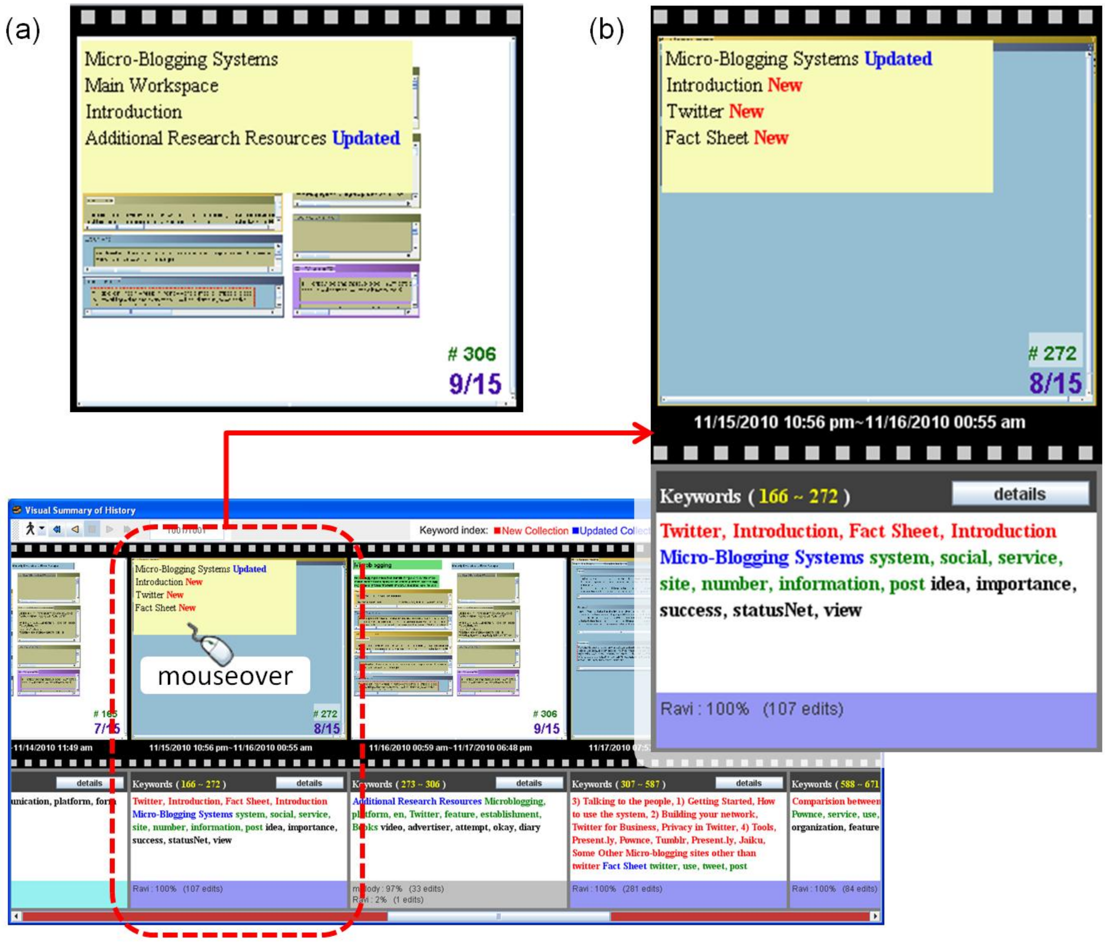


Figure 28. Visual cues in filmstrip visualization

Characterizing keywords using visual techniques is beneficial to understand the work history in VKB. Since users often take advantage of the collection hierarchy to classify information, collection-specific information can aid rapid location of activity and trace its development with the help of the visual hints. The keywords for objects are expected to be useful in that they inform users of the more particular topics involved in the activity.

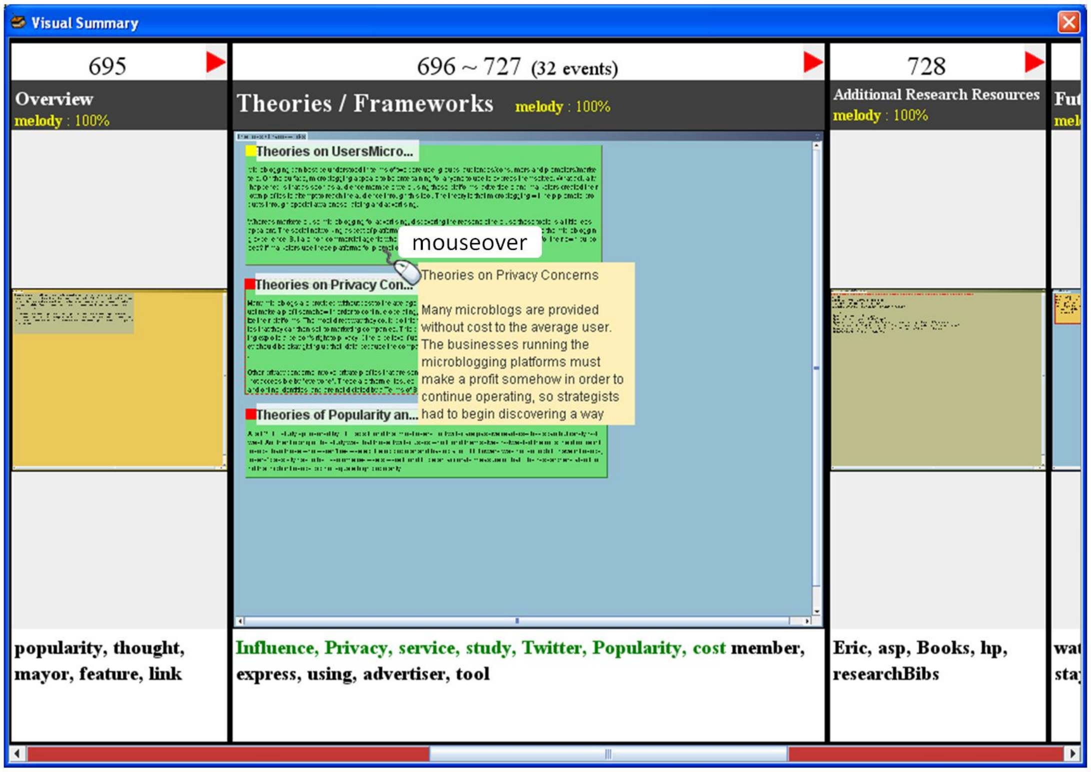
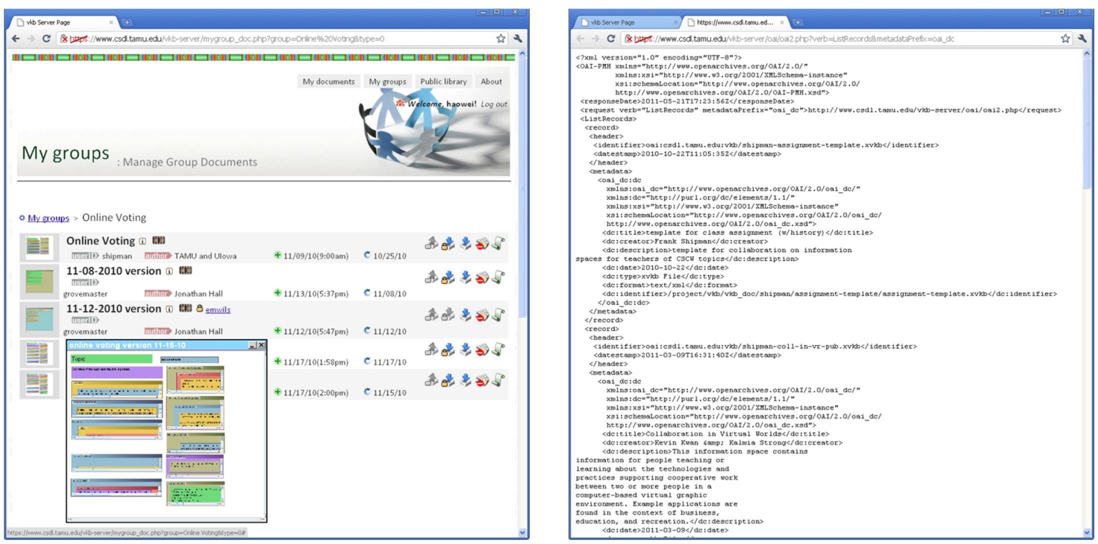


Figure 29. Visual cues in visual summary

The visual summary is improved to visually provide user activities on VKB's workspace. In Figure 29, each thumbnail includes visual cues indicating user activities that happened during the period. In the figure, the thumbnail representing the group from the 696th event to the 727th includes three dots on the left-top side of each three objects and one popup window on the middle object. The dot indicates that a certain user(s) worked on the corresponding object/collection. It is in yellow when its object or collection's is updated and red when it is created. Next to the dot, a brief text field provides the first part of textual content. For more detail, placing the mouse pointer over the text results in popup window showing more of the text. A smaller thumbnail without visual hints is provided for individual events.

7.3 Integration in VKB Server

VKB server has been developed as part of the Ensemble Pathway for computing education, one of the NSF NSDL Pathways collecting domain-specific content for education. Ensemble's goal is to establish a distributed portal that provides access to the existing collections and communities of computing educational resources. It also provides a set of tools to help educators create collections of resources.



(a) (b)

Figure 30. VKB server

VKB is employed in the project as a tool that helps teachers and students collect and organize educational resources. Traditionally, collaborations in VKB have been much like collaborations in other single-user applications like Microsoft Word – one person takes a turn editing the content and emails the resulting document to their collaborators. VKB server was developed to make such collaborations easier and to enable instructors to watch the collaboration as it proceeds.

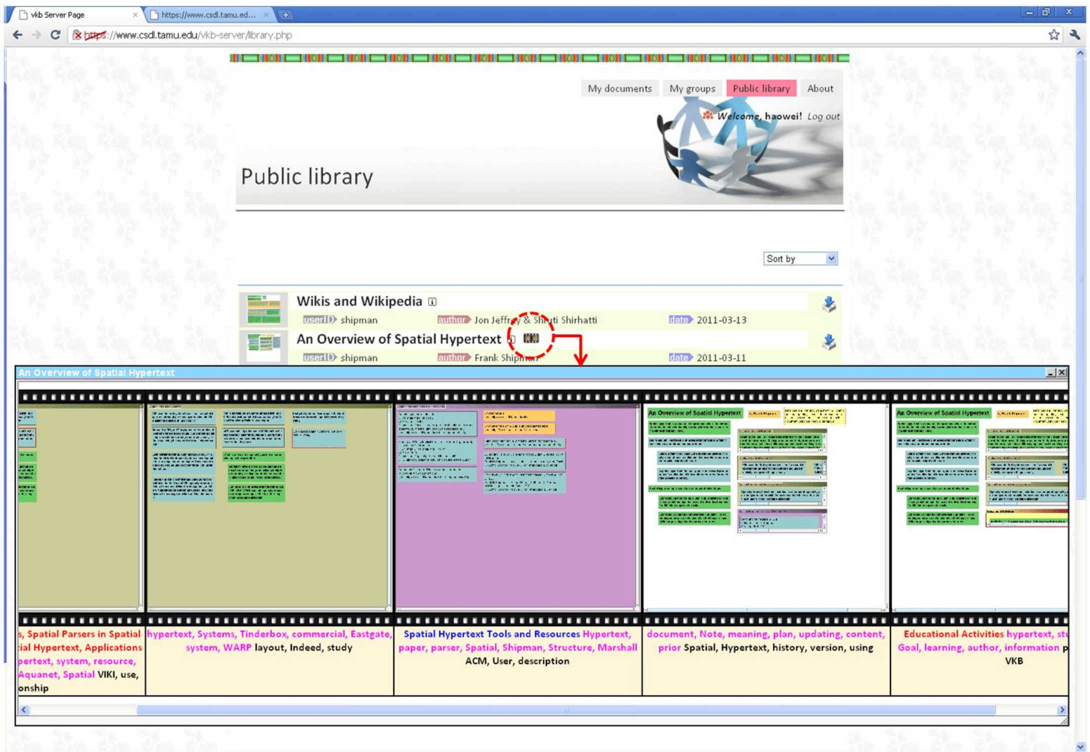


Figure 31. Filmstrip visualization in VKB server

VKB server is a service for forming project teams and sharing VKB files. As work is performed in the VKB client (a slightly augmented version of VKB 2), users can upload the VKB document to the VKB server for sharing and publishing (see Figure 30.a). The server allows them to create/join a group for collaborative editing. Once the editing is done, they can publish the resources via OAI-PMH (Open Archives Initiative

Protocol for Metadata Harvesting)[2] in the server (see Figure 30.b). This allows its content to be mined for inclusion in the Ensemble collection.

CoActEVE was integrated in the server to help users understand the progress of their collaborative work. When a document is uploaded, VKB server uses CoActIVE to generate a thumbnail visualization of the document's history (see Figure 31). It monitors the repository of the server at fixed time intervals (currently 10 minutes) to identify newly uploaded or updated VKB documents. Through the visualization, users can hopefully better understand the development of their collaborative work. They can also become aware of the work done by their collaborators while they were absent.

7.4 Integration in Design Exploration

Design Exploration (DE) Environment (see Figure 32) is a tool to gather and analyze feedback from end users to enhance software design [Moore 2007]. Through DE's construction kit, users construct rough interfaces augmented with textual argumentation. The kit looks like a graphical user interface (GUI) builder, and the users are supposed to build their own application mockup by creating a window widget and placing widgets, such as buttons and labels, in the window. Once end users create partial designs, another tool, the DE Analyzer supports software designers in analyzing the user-created designs.

DE's pre-existing history mechanism records a complete history, but it provides only one-step undo/redo operations. In order to enhance the visualizations and navigation of history, CoActIVE was integrated. DE only fulfills the minimum

[2] Open Archives Initiative Protocol for Metadata Harvesting, http://www.openarchives.org/pmh/

requirements for integration; it was to define user activity as system events and their undo/redo operations. Unlike the VKB's event, diverse event classes were created by extending the HIMEEventStruct class based on event types. The undo/redo operations were also defined in corresponding event classes. For example, the GRCEventCreate class deals with the user activity of creating a widget. This class includes the redo method for adding the widget and the undo method for deleting it.

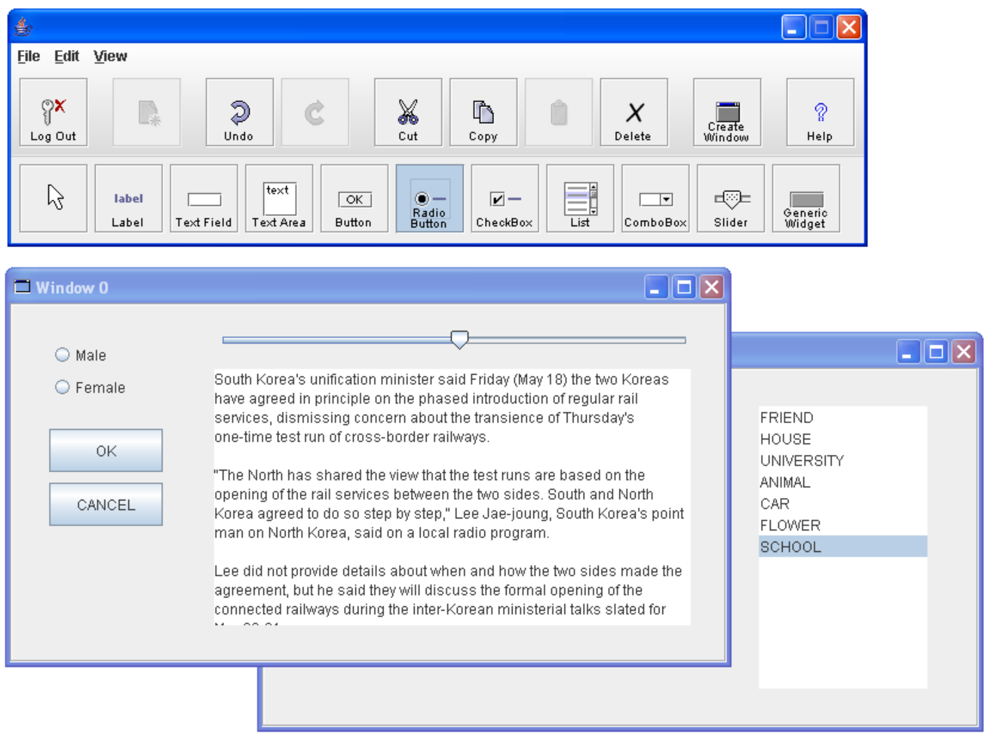


Figure 32. Design Exploration

After event classes were defined, DE employed the established components of CoActIVE. CoActIVE's history manager was adopted as it was since undo/redo operations were already defined in individual event classes. DE uses the default distance function f, time gap, to calculate similarity between events during clustering. DE also employed basic visualization components without customization.

DE provides an example of integration in an existing JAVA application with relatively minimal history capabilities. In particular, DE only had single-level undo and redo prior to CoActIVE integration although it maintained a log file with all the events for data collection purposes.

CHAPTER VIII

EVALUATION

A study was performed to explore the effect of CoActIVE's history interpretation and visualization components on users' ability to locate and comprehend the activity of others in collaborative activities. The study used previously recorded collaborations where the participants would have no prior knowledge of the activity or the people involved in the activity, as is true for someone just beginning to participate in an existing on-line collaboration, such as suggesting edits to a wiki page.

8.1 Participants

24 graduate students were recruited via email and in-person contact (see Appendix A). 75% of the participants had an engineering background and the others were from diverse areas (2 from architecture, 1 from natural science, 1 from social science, and 2 from education). The participants' ages ranged from 26 to 45. 2 female and 22 male students were participated. The study was conducted in the Center for the Study of Digital Libraries at Texas A&M University. We asked participants about their use of computers, history and collaborative applications.

All participants reported using a computer daily and have used computers for more than four years. Everyone was familiar with Windows, 12 of them could operate Unix/Linux systems, and 5 of them used Mac OS. In addition, 10 participants were used to mobile computing systems.

All participants also answered that they had experience with history tools, and have used undo/redo in document or picture editing applications. Other uses of history reported by participants included revisiting web pages, version control, and history support in an Integrated Development Environment. Among the participants, 11 people used the history tools every time, another 11 people used them as necessary, and the other 2 used them rarely. Overall, most participants felt they needed history tools for undo/redo and were satisfied with the existing tools (25% are neutral and 67% are satisfied).

15 of the 24 participants (62.5%) had experience working with other people via collaboration applications. Only two participants reported using history information to find particular activity or to understand work progress in such a collaboration. Most participants mentioned that they had a hard time recognizing the work of collaborators, and some reported not trying to understand the work of others due to such difficulties. Some participants reported browsing the modified parts of a work (e.g. document, design) relying on their memory of previous work to locate changes. Email and annotations were the most commonly reported methods to coordinate the collaboration. Subversion and web storage (e.g. Dropbox) were also used for maintaining group work.

8.2 Experimental Design

Participants were placed in the role of a teaching assistant examining collaboratively authored VKB documents to determine which students did what (see Appendix B). Each participant was asked to answer an equivalent series of questions about four different recorded activities (see Appendix C). The recorded activities (documents) were collaboratively authored by teams of two to three students spread across two classes at different universities. The four documents used were selected based on having relatively similar work practices and approximately the same amount of recorded activity. Table IV lists the topics of the four VKB spaces and the number of user events recorded for each activity.

Table IV. Selected documents for evaluation

Topic	Event count
Online Balkanization	963
Blogging	981
Online Voting	945
Microblogging	1003

As a teaching assistant, participants were asked to understand how given documents had been developed and in what way each project member had contributed to the documents. Four visualizations were provided to answer the questions; they were the history session viewer, the history interpretation viewer, the filmstrip visualization with textual summaries, and the filmstrip visualization with visual summaries.

Each participant used a different visualization to answer the questions for each recorded activity. The assignment of recorded activity to visualization type was balanced (6 for each pair). Due to the potential for learning effects, the order of the visualization was balanced across subjects, with each of the 24 participants having a unique visualization order.

A tutorial session was given before beginning the first task. The session covered the use of both VKB and the history mechanism. After the task, participants were provided a follow-up questionnaire that asked about their satisfaction with history support and visualization settings. They also were asked to compare visualization settings and provide their opinion on the user study. A brief interview followed after the survey.

8.2.1 Task

Each task included five types of questions (see Table V) to be answered by using the history interface provided (see Appendix D). Questions of type one though four were assessed based on both the time required to answer each question and the correctness of the answer. The last question was measured based on the number of correct and incorrect answers provided in the time provided (3 minutes).

Table V. Five types of questions

Type	Question
1	Find the event ID where **Jill** created the information object contained below.
2	What is the previous title of the “**How to Start a Blog**” collection?
3	Which one of **X** and **Y** was created earlier in the document?
4	Find the event ID where the collection, **“CSCW without C”** is displayed on a screen as shown below.
5	Find as many places as **Jack** worked on the “**Current research**” collection as possible in 3 minutes.

The first question is to examine the effect of the visualization on locating a specific activity by a certain user. The second question determines the visualization's effect on recovering the prior state of a specified element in the document. The third question examines the visualization's effect on comprehending the order of specific activities. The forth question explores the effect of the visualization on locating a particular state of the document. The last question tests the visualization's effect on identifying all the activity of a particular user regarding specific information.

8.3 Results

While we expected the clustering and summarization would aid the location and comprehension tasks, we were not sure which of the augmented views would be of most aid to participants and which would be preferred. Here we present results indicating the effect of the visualization on time required for tasks, error rate for tasks, and user satisfaction.

Throughout the statistical analysis of time spent on each visualization setting and each question type, the Kruskal-Wallis test (nonparametric test) is employed to investigate the existence of difference between groups. The One-Way ANOVA is not

Table VI. Overall time spent on each setting

	VS1	VS2	VS3	VS4
Mean (min)	17.04	13.43	8.62	7.44
Std Dev	5.99	4.01	2.71	2.81

VS1 : History session viewer
VS2 : History interpretation viewer
VS3 : Filmstrip + Textual summary
VS4 : Filmstrip + Visual summary

considered because not all groups' population means are normally distributed, and their variances are not equal. As a post-hoc test, the Tukey's test is used to examine which groups are significantly different from one another.

8.3.1 Time Spent on Each Visualization Setting

The mean time which 24 participants spent on the four tasks was 58.52 minutes. The total time for the evaluation was around two hours including the demographic and domain surveys, tutorial session, questionnaire, and interview.

Table VI shows the mean and standard deviation of total time the participants spent on the first four questions with each visualization. Participants took the most time

Table VII. Kruskal-Wallis test on time spent on each setting

Ranks

	Settings	N	Mean Rank
Time_spent	VS1	24	74.33
	VS2	24	62.19
	VS3	24	32.08
	VS4	24	25.40
	Total	96	

Test Statistics[a,b]

	Time_spent
Chi-Square	51.281
df	3
Asymp. Sig	.000

a : Kruskal Wallis Test

b : Grouping Variable : Time_spent

when using the history session viewer (VS1: Visual Setting #1) and the least time when using the filmstrip visualization+visual summary (VS4).

Table VIII. Tukey's test on time spent on each setting

Multiple Comparisons

Dependent Variable: Rank of TIME_SPENT
Tukey HSD

(I) VS	(J) VS	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Upper Bound	Lower Bound
1	2	12.14583	5.543354	.133	-2.35899	26.65066
	3	42.25000*	5.543354	.000	27.74518	56.75482
	4	48.93750*	5.543354	.000	34.43268	63.44232
2	1	-12.14583	5.543354	.133	-26.65066	2.35899
	3	30.10417*	5.543354	.000	15.59934	44.60899
	4	36.79167*	5.543354	.000	22.28684	51.29649
3	1	-42.25000*	5.543354	.000	-56.75482	-27.74518
	2	-30.10417*	5.543354	.000	-44.60899	-15.59934
	4	6.68750	5.543354	.624	-7.81732	21.19232
4	1	-48.93750*	5.543354	.000	-63.44232	-34.43268
	2	-36.79167*	5.543354	.000	-51.29649	-22.28684
	3	-6.68750	5.543354	.624	-21.19232	7.81732

Based on observed means.
* The mean difference is significant at the .05 level.

Table VII is the result of the Kruskal-wallis test on time spent on each setting. It shows that the difference between VS1, VS2, VS3, and VS4 is significant ($p<.001$). A post-hoc Tukey's test shows significant differences between all permutations of the pair VS3 and VS4 with the pair VS1 and VS2 with $p<.001$, The difference between VS3 and

VS4 is not significant (p=.62) nor is the difference between VS1 and VS2 (p=.13) (see Table VIII).

These results indicate that the Filmstrip Viewer enabled more efficient use of the history for answering the variety of questions covered in the overall procedure, although there was not a significant difference between the visual summary and textual summary versions.

8.3.2 Time Spent on Each Type of Question

Table IX shows both the mean and the standard deviation of time taken to perform each of the first four types of questions under four visualization settings. The results found in the overall time continue for each of the four question types. VS4 has the shortest mean time to complete each type of questions, with VS3 taking slightly longer and VS2 being somewhat faster than VS1. The notable exception was that for Type 2 questions, VS2 was only slightly slower than VS3 and took less than half the time of VS1.

Table IX. Time spent on each question type

	Means (min)				**Standard Deviation**			
	VS1	**VS2**	**VS3**	**VS4**	**VS1**	**VS2**	**VS3**	**VS4**
Type 1	4.16	3.70	2.55	2.02	1.44	2.34	1.31	0.96
Type 2	3.56	1.62	1.49	1.25	2.41	0.83	0.88	0.48
Type 3	5.23	4.21	2.59	2.46	3.19	2.24	2.05	1.65
Type 4	4.09	3.90	1.98	1.70	2.69	2.04	1.10	0.94

Table X. Kruskal-Wallis test on time spent on each type of question

- **Type 1**

Ranks

	Settings	N	Mean Rank
Time_spent	VS1	24	67.44
	VS2	24	53.27
	VS3	24	41.17
	VS4	24	32.13
	Total	96	

Test Statistics[a,b]

	Time_spent
Chi-Square	21.755
df	3
Asymp. Sig	.000

- **Type 2**

Ranks

	Settings	N	Mean Rank
Time_spent	VS1	24	72.69
	VS2	24	46.04
	VS3	24	38.08
	VS4	24	37.19
	Total	96	

Test Statistics[a,b]

	Time_spent
Chi-Square	25.599
df	3
Asymp. Sig	.000

- **Type 3**

Ranks

	Settings	N	Mean Rank
Time_spent	VS1	24	65.38
	VS2	24	58.63
	VS3	24	35.38
	VS4	24	34.63
	Total	96	

Test Statistics[a,b]

	Time_spent
Chi-Square	23.262
df	3
Asymp. Sig	.000

- **Type 4**

Ranks

	Settings	N	Mean Rank
Time_spent	VS1	24	63.19
	VS2	24	64.52
	VS3	24	35.54
	VS4	24	30.75
	Total	96	

Test Statistics[a,b]

	Time_spent
Chi-Square	29.549
df	3
Asymp. Sig	.000

a Kruskal Wallis Test / b Grouping Variable: Time_spent

The Kruskal-Wallis test shows that, for each question type, VS1, VS2, VS3, and VS4 results are significantly different (p<.001) (see Table X). Tukey's test shows a similar pattern as the overall results – that VS1 is significantly different from VS3 and VS4 for all question types (p<.003 for closest) and that VS3 and VS4 are not significantly different for any of the question types.

Table XI. Tukey's test on time spent on type 1 question

Multiple Comparisons

Dependent Variable: Rank of TIME_SPENT
Tukey HSD

(I) VS	(J) VS	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Upper Bound	Lower Bound
1	2	14.16667	7.174779	.205	-4.60697	32.94030
	3	26.27083*	7.174779	.002	7.49720	45.04447
	4	35.31250*	7.174779	.000	16.53887	54.08613
2	1	-14.16667	7.174779	.205	-32.94030	4.60697
	3	12.10417	7.174779	.336	-6.66947	30.87780
	4	21.14583*	7.174779	.021	2.37220	39.91947
3	1	-26.27083*	7.174779	.002	-45.04447	-7.49720
	2	-12.10417	7.174779	.336	-30.87780	6.66947
	4	9.04167	7.174779	.590	-9.73197	27.81530
4	1	-35.31250*	7.174779	.000	-54.08613	-16.53887
	2	-21.14583*	7.174779	.021	-39.91947	-2.37220
	3	-9.04167	7.174779	.590	-27.81530	9.73197

Based on observed means.
* The mean difference is significant at the .05 level.

Table XI is the result of Tukey's test for the type 1 question. As a reminder, the type 1 question asked participants to find when a particular user made a particular edit. From the result, we can conclude that for locating a specific activity of a specific user, VS4 is more effective than VS1 and VS2, and VS3 is more effective than VS1. There is

Table XII. Tukey's test on time spent on type 2 question

Multiple Comparisons

Dependent Variable: Rank of TIME_SPENT
Tukey HSD

(I) VS	(J) VS	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Upper Bound	Lower Bound
1	2	26.64583*	6.983781	.001	8.37197	44.91970
	3	34.60417*	6.983781	.000	16.33030	52.87803
	4	35.50000*	6.983781	.000	17.22613	53.77387
2	1	-26.64583*	6.983781	.001	-44.91970	-8.37197
	3	7.95833	6.983781	.666	-10.31553	26.23220
	4	8.85417	6.983781	.586	-9.41970	27.12803
3	1	-34.60417*	6.983781	.000	-52.87803	-16.33030
	2	-7.95833	6.983781	.666	-26.23220	10.31553
	4	.89583	6.983781	.999	-17.37803	19.16970
4	1	-35.50000*	6.983781	.000	-53.77387	-17.22613
	2	-8.85417	6.983781	.586	-27.12803	9.41970
	3	-.89583	6.983781	.999	-19.16970	17.37803

Based on observed means.
* The mean difference is significant at the .05 level.

no significant difference between VS3 and VS4 (p=.590), VS2 and VS3 (p=.336), and VS1 and VS2 (p=.205).

The type 2 question is to investigate which visualization setting performs more effectively on recovering the previous state of a particular element in a document.

Table XIII. Tukey's test on time spent on type 3 question

Multiple Comparisons

Dependent Variable: Rank of TIME_SPENT
Tukey HSD

(I) VS	(J) VS	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Upper Bound	Lower Bound
1	2	6.75000	7.100756	.778	-11.82995	25.32995
	3	30.00000*	7.100756	.000	11.42005	48.57995
	4	30.75000*	7.100756	.000	12.17005	49.32995
2	1	-6.75000	7.100756	.778	-25.32995	11.82995
	3	23.25000*	7.100756	.008	4.67005	41.82995
	4	24.00000*	7.100756	.006	5.42005	42.57995
3	1	-30.00000*	7.100756	.000	-48.57995	-11.42005
	2	-23.25000*	7.100756	.008	-41.82995	-4.67005
	4	.75000	7.100756	1.000	-17.82995	19.32995
4	1	-30.75000*	7.100756	.000	-49.32995	-12.17005
	2	-24.00000*	7.100756	.006	-42.57995	-5.42005
	3	-.75000	7.100756	1.000	-19.32995	17.82995

Based on observed means.
* The mean difference is significant at the .05 level.

Concerning the time spent on the question, Tukey's test shows that participants who worked with VS2, VS3 and VS4 spent less time than those who with VS1. The difference among VS2, VS3 and VS4 is not significant (VS3 and VS4: p=.999, VS2 and VS4: p=.586, VS2 and VS3: p=.666) (see Table XII).

Table XIII shows which settings are different each other in their effectiveness on understanding the order of specific activities (type 3 question). It indicates that, VS3 and

Table XIV. Tukey's test on time spent on type 4 question

Multiple Comparisons

Dependent Variable: Rank of TIME_SPENT
Tukey HSD

(I) VS	(J) VS	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Upper Bound	Lower Bound
1	2	-1.33333	6.782637	.997	-19.08088	16.41422
	3	27.64583*	6.782637	.001	9.89828	45.39338
	4	32.43750*	6.782637	.000	14.68995	50.18505
2	1	1.33333	6.782637	.997	-16.41422	19.08088
	3	28.97917*	6.782637	.000	11.23162	46.72672
	4	33.77083*	6.782637	.000	16.02328	51.51838
3	1	-27.64583*	6.782637	.001	-45.39338	-9.89828
	2	-28.97917*	6.782637	.000	-46.72672	-11.23162
	4	4.79167	6.782637	.894	-12.95588	22.53922
4	1	-32.43750*	6.782637	.000	-50.18505	-14.68995
	2	-33.77083*	6.782637	.000	-51.51838	-16.02328
	3	-4.79167	6.782637	.894	-22.53922	12.95588

Based on observed means.
* The mean difference is significant at the .05 level.

VS4 are more effective than VS1 and VS2. The difference between VS3 and VS4 is not significant (p=1.000) nor is the difference between VS1 and VS2 (p=.778).

For the task to compare the effectiveness of the visualization on locating a particular visual state of the document, Tukey's test (see Table XIV) concludes that VS3 and VS4 are more effective than VS1 and VS2. The difference between VS3 and VS4 is not significant (p=.894) nor is the difference between VS1 and VS2 (p=.997).

8.3.3 Correctness of Answers

Answers provided by participants were not always correct. Table XV shows the number of incorrect answers for each visualization for each of the first four question types. Among the settings, 11 errors (55% of the total errors across all conditions) were found when using the history session viewer. From this result, it seems that working with low-level history yields a higher error rate. Particularly, participants seemed to have difficulty in understanding the order of specific activities (type 3) and locating a particular state of the document in the history (type 4). The history interpretation viewer also resulted in a number of errors when locating a particular state of the document.

Table XV. Number of incorrect answers

	VS1	VS2	VS3	VS4	Total
Type 1	1	0	2	0	3
Type 2	0	0	0	0	0
Type 3	4	1	0	0	5
Type 4	6	5	0	1	12
Total	11	6	2	1	20

Table XVI. Number of correct answers for type 5 question

	VS1	VS2	VS3	VS4
Mean	1.83	1.58	2.38	4.42
Std Dev	1.37	1.47	0.92	1.25

The fifth question in each task was designed to assess the efficiency of each interface for finding all the activity of a particular type. Table XVI presents the average number of correct answers and their standard deviation in the three minutes provided.

VS2 was expected to be better than VS1 due to its support for history interpretation. However, many participants reported felling uncomfortable when

Table XVII. Kruskal-Wallis test on type 5 question on each setting

Ranks

	Settings	N	Mean Rank
Correct_ans	VS1	24	36.71
	VS2	24	32.88
	VS3	24	46.21
	VS4	24	78.21
	Total	96	

Test Statistics[a,b]

	Correct_ans
Chi-Square	40.849
df	3
Asymp. Sig	.000

a : Kruskal Wallis Test

b : Grouping Variable : Correct_ans

investigating the structure of a clustering tree and a few participants had difficulty in matching keywords with target activity. For this reason, they tended to use VKB's embedded history playback interface more (to rewind and fast-forward through the history). This resulted in similar mean times between VS1 and VS2 with VS1 being slightly better.

Table XVIII. Tukey's test on type 5 question

Multiple Comparisons

Dependent Variable: Rank of CORRECT_ANS
Tukey HSD

(I) VS	(J) VS	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Upper Bound	Lower Bound
1	2	3.83333	6.052121	.921	-12.00274	19.66940
	3	-9.50000	6.052121	.401	-25.33607	6.33607
	4	-41.50000*	6.052121	.000	-57.33607	-25.66393
2	1	-3.83333	6.052121	.921	-19.66940	12.00274
	3	-13.33333	6.052121	.130	-29.16940	2.50274
	4	-45.33333*	6.052121	.000	-61.16940	-29.49726
3	1	9.50000	6.052121	.401	-6.33607	25.33607
	2	13.33333	6.052121	.130	-2.50274	29.16940
	4	-32.00000*	6.052121	.000	-47.83607	-16.16393
4	1	41.50000*	6.052121	.000	25.66393	57.33607
	2	45.33333*	6.052121	.000	29.49726	61.16940
	3	32.00000*	6.052121	.000	16.16393	47.83607

Based on observed means.
* The mean difference is significant at the .05 level.

Here again the filmstrip with visual summaries performed the best. The Kruskal-Wallis test shows that the difference between VS1, VS2, VS3, and VS4 is significant ($p<.001$) (see Table XVII). This time the difference between this interface and all other interfaces is significant (VS3 and VS4: $p=.000$, VS2 and VS4: $p=.000$, VS1 and VS4: $p=.000$) (see Table XVIII).

8.3.4 Questionnaire Results

After the completion of the task, participants responded to a set of questions regarding their satisfaction. The questions employed a Likert scale where a value of 1 indicated strong disagreement, and 7 indicated strong agreement (see Appendix E).

Overall, participants' reported preference matched the performance results: VS4 (mean=6.13, std dev=0.88), VS3 (mean=4.96, std dev=0.82), VS2 (mean=3.96, std dev=1.37), and VS1 (mean=2.21, std dev=1.61). Among visualization techniques, the visual summary was the most preferred (mean=6.21, std dev=0.91), the thumbnail visualization technique was somewhat less preferred (mean=5.79, std dev=1.25), and the textual summary (mean=4.96, std dev=1.62) was less still although still above the neutral rating of 4. As noted above, the rating for VS1, which is most similar to the history interface found in VKB 2, indicated a strong displeasure among participants (mean 2.21, std dev=1.61).

Participants liked to explore the automatically-generated hierarchic interpretation of history (mean=6.08, std dev=0.90). The keywords (mean=5.79, std dev=0.90) and author information (mean=6.13, std dev=0.73) provided with history clustering were also found valuable.

Among the four settings, VS4 was selected as the most intuitive (17 participants) to perform the task, the most useful for user orientation (16 participants) and the best for finding target information in history (19 participants).

8.3.5 Interviews

Participants were interviewed on how they used four visualization settings to perform the given task. Most of them mentioned that the task under VS1 was difficult to perform. This matches the fact that they spent more time with the history session viewer to answer the questions than the other interfaces. They also frequently had to depend on the history playback interfaces to augment the information in the history session viewer. Two people handled the interface well, and especially, one participant liked it the most. However, they spent the most time on the task, and they generated more incorrect answers than other participants.

In most cases for VS2, participants utilized the summary in the history interpretation viewer to understand past activity and narrowed the scope of investigation by navigating down the tree of the viewer. But, one participant picked VS2 as the most difficult, because he/she was not familiar with the tree representation. A few participants reported being disoriented when their interpretation of certain activity based on the information in the viewer was not valid to answer the questions in the task.

The text summary component of VS3 was liked by most participants. They could confirm their interpretation by comparing the textual content of the component with the keywords of the filmstrip visualization. A couple of participants mentioned that the filmstrip visualization did not provide enough information since the presentation of snapshots and summary was too coarse to understand the progress of work in history. However, the addition of the visual summary component of VS4 settled this problem and 23 people were satisfied with it. One participant was not accustomed to the visual summary and chose the history interpretation viewer as the most preferred interface.

8.4 Summary

Overall, the results of the study show a performance advantage and a preference for the filmstrip visualization with visual summaries. All interfaces providing access to the automatically clustered history were preferred over the traditional list view of VS1.

CHAPTER IX

CONCLUSIONS AND FUTURE WORK

The increase in server-based and cloud-based applications has brought a corresponding increase in long-term collaborations among people who may never know one another. Records of user activity support can be used to provide an understanding of prior effort in such situations but locating and comprehending particular activity within large history records can be difficult for users. CoActIVE is a history mechanism that clusters system-level activity into higher-level episodes, generates textual and visual summaries of these episodes, and provides a variety of history visualizations based on the inferred episodes. Additionally, CoActIVE supports branching history, with the understanding that asynchronous authoring and design tasks often involve the parallel development of alternatives. CoActIVE can be adapted for use with most Java applications that already support undo/redo.

An evaluation compared performance and satisfaction for participants answering questions that required them to locate particular events or states in a recorded history and to comprehend who performed different activities and the order of activity. Two list/tree views of the history record and two filmstrip views of the history record were compared. The list views included one more typical event-list view as found in current applications and an augmented tree view of the hierarchically-clustered activity. The two filmstrip views differed in that one provided textual summaries of changes and one provided visual summaries. The results of the study showed significant improvements for all three new interfaces over the traditional event view. Participants were able to perform the tasks most efficiently with the fewest errors with the filmstrip view with visual summaries. This view was also participants' favored view in post-task surveys and interviews.

Future work can build on these results. The evaluation performed did not specifically examine the quality of the clustering results. Comparing alternative event-clustering algorithms has the potential to improve the overall support of such interfaces. Similarly, the four visualizations compared are examples from a large design space for history visualization. The results show visual presentations of history and state were the most valuable but that may differ depending on the application being supported. Finally, while we have integrated CoActIVE in a multiple Java applications, it is future work to determine if the results seen here will transfer to applications that are less spatially expressive.

REFERENCES

BAUDISCH, P. AND ROSENHOLTZ, R. 2003. Halo: A technique for visualizing off-screen objects. In *Proceedings of the 22nd ACM SIGCHI Conference on Human Factors in Computing Systems*, Ft. Lauderdale, FL, ACM, 481-488.

BEGOLE, J.B., TANG, J.C., SMITH, R.B., AND YANKELOVICH, N. 2002. Work rhythms: Analyzing visualizations of awareness histories of distributed groups. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, New Orleans, LA, ACM, 334-343.

BERLAGE, T. 1994. A selective undo mechanism for graphical user interfaces based on command objects. *ACM Trans. Comput.-Hum. Interact. 1*, 269-294.

BUSH, V. 1945. As we may think. *The Atlantic Monthly 176*, 101-108.

CHENGZHENG, S. 2000. Undo any operation at any time in group editors. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, Philadelphia, PA, ACM, 191-200.

DERTHICK, M. AND ROTH, S.F. 2000. Data exploration across temporal contexts. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*, New Orleans, LA, ACM, 60-67.

HILL, W.C., HOLLAN, J.D., WROBLEWSKI, D., AND MCCANDLESS, T. 1992. Edit wear and read wear. In *Proceedings of the 11th ACM SIGCHI Conference on Human Factors in Computing Systems*, Monterey, CA, ACM, 3-9.

KLEMMER, S.R., THOMSEN, M., PHELPS-GOODMAN, E., LEE, R., AND LANDAY, J.A. 2002. Where do web sites come from? capturing and interacting with design history. In *Proceedings of the 21st ACM SIGCHI Conference on Human Factors in Computing Systems*, Minneapolis, MN, ACM, 1-8.

LEE, A. 1992. Investigations into history tools for user support. Ph.D. dissertation, *Department of Computer Science*, University of Toronto.

MILIC-FRAYLING, N., JONES, R., RODDEN, K., SMYTH, G., BLACKWELL, A., AND SOMMERER, R. 2004. Smartback: Supporting users in back navigation. In *Proceedings of the 13th International Conference on World Wide Web*, New York, ACM, 63-71.

MOORE, J.M. 2007. Design Exploration: Engaging a larger user population. Ph.D. dissertation, *Department of Computer Science and Engineering*, Texas A&M University.

NAKAMURA, T. AND IGARASHI, T. 2008. An application-independent system for visualizing user operation history. In *Proceedings of the 21st ACM Symposium on User Interface Software and Technology*, Monterey, CA, ACM, 23-32.

NGUYEN, T.N., MUNSON, E.V., AND BOYLAND, J.T. 2004. The molhado hypertext versioning system. In *Proceedings of the 15th ACM Conference on Hypertext and Hypermedia*, Santa Cruz, CA, ACM, 185-194.

PLAISANT, C., MILASH, B., ROSE, A., WIDOFF, S., AND SHNEIDERMAN, B. 1996. LifeLines: Visualizing personal histories. In *Proceedings of the 14th ACM SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, British Columbia, Canada, ACM, 221-227.

PLAISANT, C., ROSE, A., RUBLOFF, G., SALTER, R., AND SHNEIDERMAN, B. 1999. The design of history mechanisms and their use in collaborative educational simulations. In *Proceedings of the 1999 Conference on Computer Support for Collaborative Learning*, Palo Alto, CA, International Society of the Learning Sciences, 348-359.

REEVES, B. 1993. Supporting collaborative design by embedded communication and history in design artifacts. Ph.D. dissertation, *Department of Computer Science*, University of Colorado.

ROSENBERG, J. 1996. The structure of hypertext activity. In *Proceedings of the 7th ACM Conference on Hypertext*, Bethesda, MD, ACM, 22-30.

SHIPMAN, F.M. AND HSIEH, H. 2000. Navigable history: A reader's view of writer's time. *New Review of Hypermedia and Multimedia 6*, 147-167.

SHIPMAN, F.M., HSIEH, H., MALOOR, P., AND MOORE, M. 2001. The visual knowledge builder: A second generation spatial hypertext. In *Proceedings of the 12th ACM Conference on Hypertext and Hypermedia*, Arhus, Denmark, ACM, 113-122.

SHIRAI, Y., YAMAMOTO, Y. AND NAKAKOJI, K. 2009. Time-based authoring tools for informal information management. In *Proceedings of the 2nd Symposium on Interactive Visual Information Collections and Activity*, Austin, TX, 23-28.

SUCHMAN, L. 1987. *Plans and Situated Actions: The Problem of Human-machine Communication*. Cambridge University Press, New York.

TERRY, M., MYNATT, E.D., NAKAKOJI, K., AND YAMAMOTO, Y. 2004. Variation in element and action: Supporting simultaneous development of alternative solutions. In *Proceedings of the 23rd ACM SIGCHI Conference on Human Factors in Computing Systems*, Vienna, Austria, ACM, 711-718.

VIGAS, F.B., WATTENBERG, M., AND DAVE, K. 2004. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the 23rd ACM SIGCHI Conference on Human Factors in Computing Systems*, Vienna, Austria, ACM, 575-582.

WEXELBLAT, A. AND MAES, P. 1999. Footprints: history-rich tools for information foraging. In *Proceedings of the 18th ACM SIGCHI Conference on Human Factors in Computing Systems*, Pittsburgh, PA, ACM, 270-277.

ZELLWEGER, P.T., MACKINLAY, J.D., GOOD, L., STEFIK, M., AND BAUDISCH, P. 2003. City lights: Contextual views in minimal space. In *Proceedings of the 22nd ACM CHI '03 Extended Abstracts on Human Factors in Computing Systems*, Ft. Lauderdale, FL, ACM, 838-839.

APPENDIX A

DEMOGRAPHICS AND DOMAIN SURVEY SHEET

Demographics & Domain Survey

Evaluation for Comprehension and Navigation of User History via a New History Mechanism

Subject ID: ______________________ **Date: ______________________**

Instructions:

- ✓ **Please fill in / circle value(s) or use an X to indicate your response.**
- ✓ **Please do not answer a question if it makes you uncomfortable or you would not like to answer it for any other reason.**

<u>Personal Information</u>

1. Age group
   a) 18 ~ 25
   b) 26 ~ 35
   c) 36 ~ 45
   d) 46 ~ 60

2. Gender
   a) Male
   b) Female

3. Academic background
   a) Engineering
   b) Pure Sciences
   c) Social Sciences
   d) Architecture
   e) Business
   f) Education
   g) Other (Please Specify): ___________________

Computer Experience

4. How long have you used computers?
   a) Less than a year
   b) A year to two years
   c) Two years to four years
   d) More than four years
   e) No experience

5. How often do you use a computer?
   a) Daily (almost every day)
   b) Weekly (2 to 3 times a week)
   c) Monthly (2 to 3 times a month)
   d) Less than once a month

6. What type of OS do you use? (please circle all that apply)
   a) Windows (Microsoft)
   b) UNIX / LINUX
   c) Mac OS
   d) Mobile OS (Android, iOS, etc)
   e) Other (Please Specify): ___________________

Experience with History Tools

Many applications provide history tools that allow users navigate back to the previous state of their work. For example, you can undo the edits that you made in document / photo editing applications such as MS office and Photoshop. Another example is a web browser that allows you to go back to the previously visited web page.

7. Have you ever tried a history tool in your frequently used applications?
   a) Yes.
   b) No.

8. How frequently do you use a history tool?
   a) I like to use history tools and I use them every time.
   b) I use history tools if necessary.
   c) I rarely use history tools.
   d) I don’t use history tools.

9. If you have experience with history tools, could you explain in what way you have used them?

____

____

____

10. Have you followed back the steps of your previous work by using history tools?
    a) Yes.
    b) No.

11. If you answered "yes" to the previous question, could you explain the reason?

    ____

    ____

    ____

12. Have you ever worked with other people via collaboration applications such as Google Docs?
    a) Yes.
    b) No.

13. If you answered "yes" to the previous question, could you explain how you recognized other collaborator's contribution?

    ____

    ____

    ____

14. Please list the names of applications which have history tools you use.

    ____

    ____

    ____

15. How are you satisfied with history tools that currently you are using?

1	2	3	4	5	6	7
Strongly disagree			Neutral			Strongly agree

APPENDIX B

TASK SHEET

Task

Evaluation for Comprehension and Navigation of User History via a New History Mechanism

General

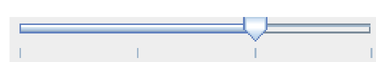
You are a teaching assistant for a CSCW (Computer Supported Collaborative Work) class. The class has offered a team project in which a couple of students create a document collaboratively by using the Visual Knowledge Builder (visual information management system). Particularly, in the team composed of more than two members, each member works asynchronously at different time and places.

Now as a teaching assistant, you are supposed to report what each group's document describes to your professor. VKB provides a history mechanism, so that you can understand how a document has been worked out. Since there are four settings in the mechanism, you have decided to try each setting and choose what you like.

Task Overview

You are supposed to understand a given document by using a history mechanism. The task consists of four sub-tasks and each of them provides one VKB document to work with.

Basically, the mechanism provides two navigation interfaces for all sub-tasks:

<table>
<tr><td>History player: you can play (forward/backward) past editing activities in a document by step-by-step or continuous manner.</td><td>
</td></tr>
<tr><td>History slider: you can jump to a certain time of past editing in a document by dragging a knob.</td><td></td></tr>
</table>

A distinct interface for each sub-task is also provided, and it will be introduced in an additional sub-task sheet.

Five questions are assigned to each sub-task (total 24 questions), and you need to answer the questions based on your understanding on the given document. The order of sub-tasks and a document for each sub-task are randomly assigned by an investigator.

Steps

1. Before the sub-tasks, learn how to use the Visual Knowledge Builder through the provided tutorial.

2. Once you finish the VKB tutorial, ask the investigator for four sub-task sheets.

    - Please follow the order (1~4) printed on each sub-task sheet.

Go to the next step ➔

APPENDIX C

SUB-TASK SHEETS

Sub-task (   )

Evaluation for Comprehension and Navigation of User History via a New History Mechanism

Steps

1. Investigate the history mechanism for **5 minutes**.
   a. **Double-click** a sample file at **C:\VKB_Documents\sample.xvkb**.
      (This step runs the Visual Knowledge Builder with the sample file open.)
   b. **Select** a '**Use History Session Dialog**' option in the history setting menu.
      (<u>Main menu > History > Configuration...</u>)

<u>History session dialog</u> provides **past activities at low-level** as well as their **high-level grouping** (continuously groups activities if their time gap is within 3 hours).  ← Pressing this button will show the dialog box.

   c. If you get familiar with the interfaces, **exit VKB**.

2. Open a document by double-clicking the document below.
   Location: C: \ VKB_Documents_________________.xvkb

3. **Answer five questions** assigned to the data file (**max 25 minutes**).

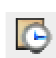
Go to the question sheet ➔

Sub-task (   )

Evaluation for Comprehension and Navigation of User History via a New History Mechanism

Steps

1. Investigate the history mechanism for **5 minutes**.
   a. **Double-click** a sample file at **C:\VKB_Documents\sample.xvkb**.
      (This step runs the Visual Knowledge Builder with the sample file open.)
   b. **Select** a '**Use History Interpretation Dialog**' option in the history setting menu.
      (Main menu > History > Configuration...)

      History interpretation dialog provides **interpretation of history** by grouping past activities into high-level activities based on the similarity.
      Keywords are also provided for the group of interpretation.
       ← Pressing this button will show the dialog box.
   c. If you get to be familiar with the interfaces, **exit VKB**.

2. Open a document by double-clicking the document below.
   Location: C: \ VKB_Documents_________________.xvkb

3. **Answer five questions** assigned to the data file (**max 25 minutes**).

Go to the question sheet →

Sub-task (   )

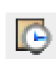
Evaluation for Comprehension and Navigation of User History via a New History Mechanism

Steps

1. Investigate the history mechanism for **5 minutes**.
   a. **Double-click** a sample file at **C:\VKB_Documents\sample.xvkb**.
      (This step runs the Visual Knowledge Builder with the sample file open.)
   b. **Select** a '**Use Filmstrip Visualization**' option in the history setting menu.
      (<u>Main menu > History > Configuration...</u>)

      Filmstrip visualization shows how past activities have progressed through the filmstrip that captures changes of a VKB workspace from the beginning to the end.

      - The filmstrip captures the workspace when there are major changes.
      - Keywords for the captured workspace are provided.

       ← Pressing this button will show the filmstrip visualization.
   c. If you get to be familiar with the interfaces, **exit VKB**.

2. Open a document by double-clicking the document below.
   Location: C: \ VKB_Documents________________.xvkb

3. **Answer five questions** assigned to the data file (**max 25 minutes**).

Go to the question sheet →

Sub-task (   )

Evaluation for Comprehension and Navigation of User History via a New History Mechanism

Steps

1. Investigate the history mechanism for **5 minutes**.
   a. **Double-click** a sample file at **C:\VKB_Documents\sample.xvkb**.
      (This step runs the Visual Knowledge Builder with the sample file open.)
   b. **Select** a '**Use Visual Summary**' option in the history setting menu.
      (Main menu > History > Configuration...)

      > **Visual summary** shows how past activities have progressed by capturing a VKB workspace when there were major visual changes.
      > - It provides visual cues indicating which parts of the workspace were changed.
      > - Keywords for the captured workspace are provided.
      >
      >  ← Pressing this button will show the visual summary.

   c. If you get to be familiar with the interfaces, **exit VKB**.

2. Open a document by double-clicking the document below.
   Location: C: \ VKB_Documents__________________.xvkb

3. **Answer five questions** assigned to the data file (**max 25 minutes**).

Go to the question sheet →

APPENDIX D

QUESTION SHEET

Question Sheet for Sub-task (   )

Evaluation for Comprehension and Navigation of User History via a New History Mechanism

Blogging.xvkb

※Before start, please enter the start time of this task: (   )

1. Find the event ID where **rhema** created the information object containing below.

> **"NOTE TO ZIMMERMAN: I'm going to leave stuff that is descriptive like a note in brackets []. We can delete it later. If you want to make a note in the context of some text with a question, just put it in brackets."**

※Please enter the time when you finish Q1: (   )

2. What is the previous title of **"How to Start a Blog"** collection?
(collection: container structure of VKB)

※Please enter the time when you finish Q2: (   )

3. Which one was created earlier in the document?

(   ) **" Spam Blogs**

Spam blogs are blogs that are not good for the general internet community. They contain automatically generated garbage made only to make ads appear more often."

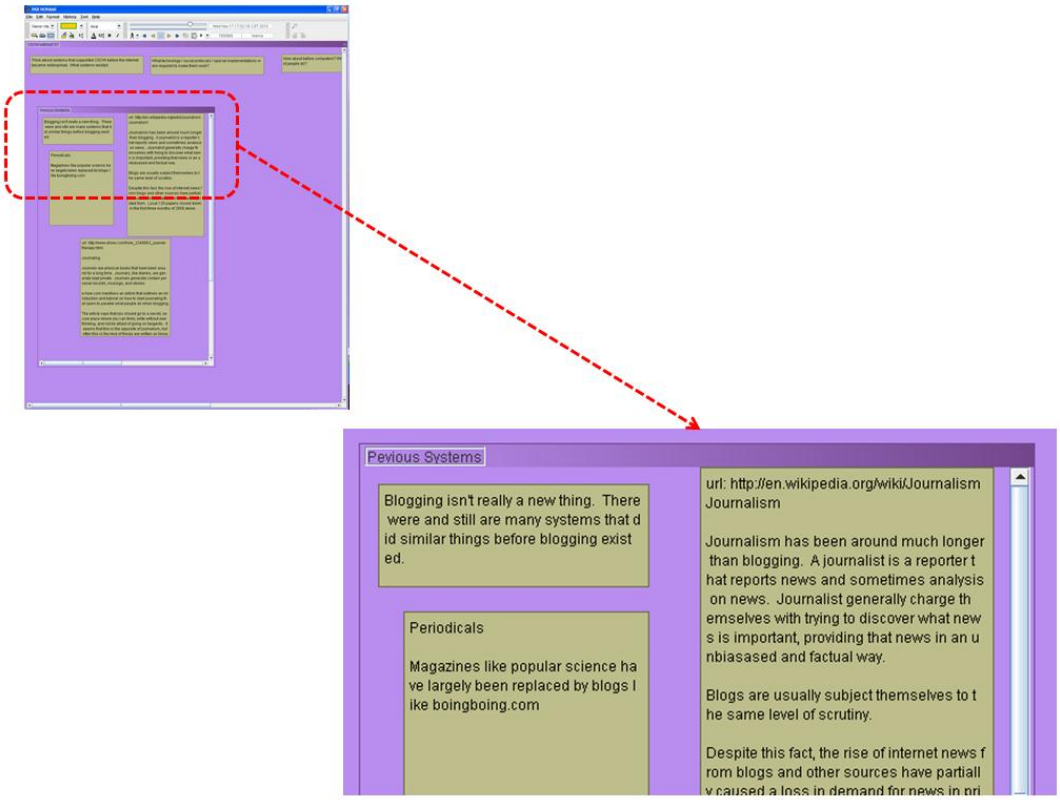
⇨ From the object in "Theories/Frameworks" collection

(   ) **"Pevious System"** - typo in the document.

⇨ Name of a collection

※Please enter the time when you finish Q3: (   )

4. Find the event ID where the collection, **" CSCW without "C" "** is displayed on a screen as shown below.



※Please enter the time when you finish Q4: (   )

5. Find as many places as **rhema** worked on **"CSCW without C"** collection. (3 mins)

From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.

(1) When you have finished this task, go to the next sub-task. ➔
(2) If you have finished all assigned tasks (total 4), ask the investigator for the next step. ➔

Question Sheet for Sub-task (   )

Evaluation for Comprehension and Navigation of User History via a New History Mechanism

Online_Voting.xvkb

※Before start, please enter the start time of this task: (   )

1. Find the event ID where **Jon** created the information object containing below.

"Langer wrote of using a 3rd-party system to handle voting. Outsourcing e-voting to a 3rd party is not widespread use."

※Please enter the time when you finish Q1: (   )

2. Find the event ID where the collection **"Pepsi Refresh Project"** was created.
   (collection: container structure of VKB)

※Please enter the time when you finish Q2: (   )

3. Which one was created earlier in the document?

(   ) **"Tagging and tag clouds are essentially a form of casual online voting in which users select among a group of associated terms in answer to a particular question, the votes are counted, and the results are reflected in the relative font size of the word as it is displayed in the aggregate? Cloud? of terms….."**

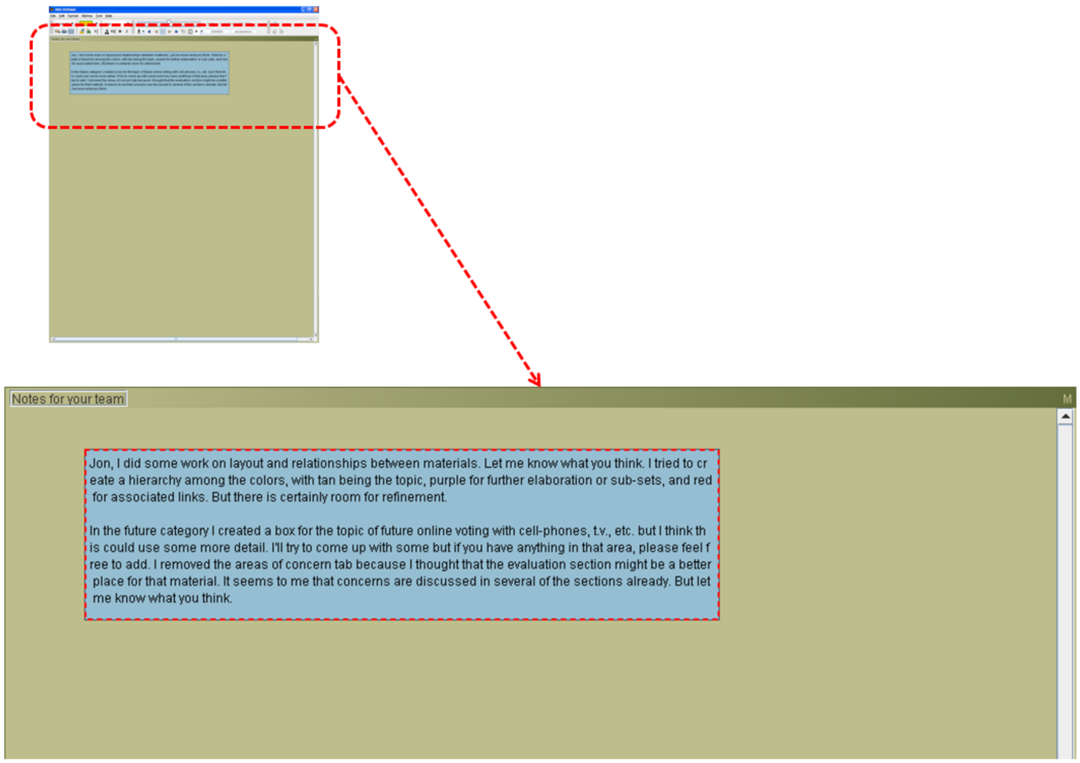
⇨ From the object in "Future Direction" collection

(   ) **"Voting Software Systems"**

⇨ Name of a collection

※Please enter the time when you finish Q3: (   )

4. Find the event ID where the collection, **"Notes for your team"** is displayed on a screen as shown below.



※Please enter the time when you finish Q4: (______________________)

5. Find as many places as **Jon** worked on **"Current research"** collection. (3 mins)

From (event ID): ______________________ To (event ID): ______________________.
From (event ID): ______________________ To (event ID): ______________________.
From (event ID): ______________________ To (event ID): ______________________.
From (event ID): ______________________ To (event ID): ______________________.
From (event ID): ______________________ To (event ID): ______________________.
From (event ID): ______________________ To (event ID): ______________________.
From (event ID): ______________________ To (event ID): ______________________.

(1) When you have finished this task, go to the next sub-task. ➔
(2) If you have finished all assigned tasks (total 4), ask the investigator for the next step. ➔

Question Sheet for Sub-task (   )

Evaluation for Comprehension and Navigation of User History via a New History Mechanism

Online_Balkanization.xvkb

※Before start, please enter the start time of this task: (   )

1. Find the event ID where **Subhadeep** created the information object containing below.

> **"Wikipedia is an unbiased resource freely available to the people so its necessary that it does not become overly balkanized. "**

※Please enter the time when you finish Q1: (   )

2. What is the previous title of **"Theories/Studies"** collection?
   (collection: container structure of VKB)

※Please enter the time when you finish Q2: (   )

3. Which one was created earlier in the document?

(   ) **".Internet has now become an integral part of our lives. We can now find more and more people involve themselves in the world of social networks, blogs, forums."**

⇨ From the object in "Definition and Significance" collection

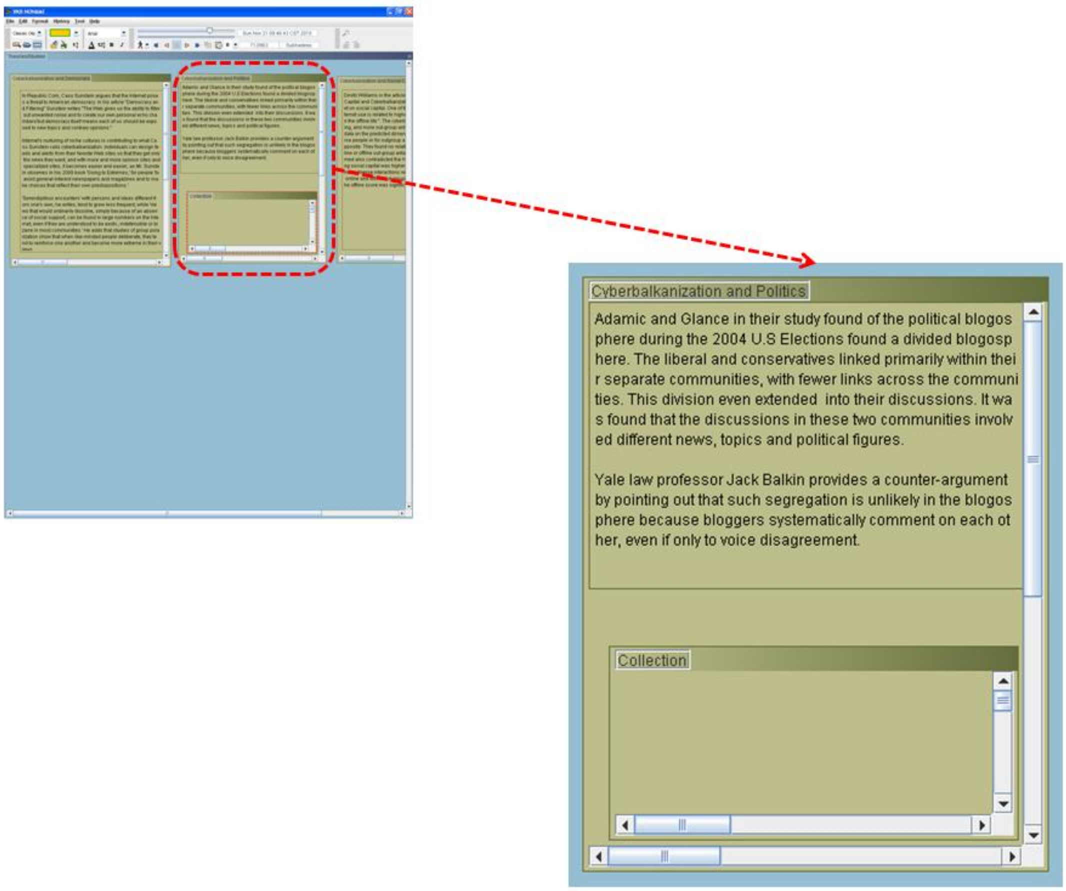
---

(   ) **"Cyberbalkanization and Democracy"**

⇨ Name of a collection

※Please enter the time when you finish Q3: (   )

4. Find the event ID where the collection, **"Cyberbalkanization and Politics"** is visually same as the below screenshot.



※Please enter the time when you finish Q4: (   )

5. Find as many places as **subhadeep** worked on **"Future Direction"** collection. (3 mins)

From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.

(1) When you have finished this task, go to the next sub-task. ➔
(2) If you have finished all assigned tasks (total 4), ask the investigator for the next step. ➔

Question Sheet for Sub-task (   )

Evaluation for Comprehension and Navigation of User History via a New History Mechanism

Microblogging.xvkb

※Before start, please enter the start time of this task: (   )

1. Find the event ID where **melody** created the information object containing below.

> **"Blogging All-in-one for Dummies (Google Books)**
> **Books VIII: Microblogging with Twitter (2010)"**

※Please enter the time when you finish Q1: (   )

2. What is the previous title of **"Traditional Communication"** collection?
   (collection: container structure of VKB)

※Please enter the time when you finish Q2: (   )

3. Which one was created earlier in the document?

(   ) **"Theories of Popularity and Influence**

A fall 2010 study sponsored by HP Labs found that most users on Twitter are passive readers--"

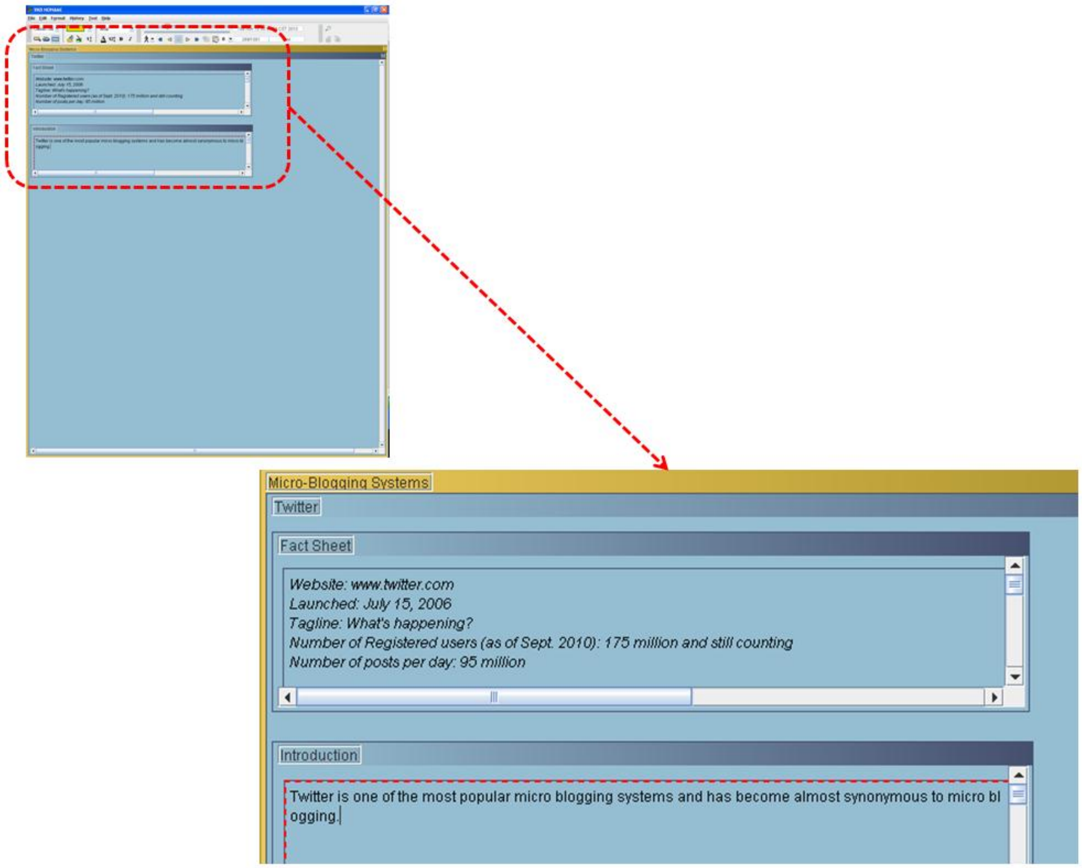
⇨ From the object in "Theories/Frameworks" collection

(   ) **"Success Factors of Micro-blogging systems"**

⇨ Name of a collection

※Please enter the time when you finish Q3: (   )

4. Find the event ID where the collection, **"Twitter"** is visually same as the below screenshot.



※Please enter the time when you finish Q4: ( )

5. Find as many places as **Ravi** worked on **"How to use the system"** collection. (3 mins)

From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.
From (event ID): ____________________ To (event ID): ____________________.

(1) When you have finished this task, go to the next sub-task. ➔
(2) If you have finished all assigned tasks (total 4), ask the investigator for the next step. ➔

APPENDIX E

QUESTIONNAIRE SHEET

Subject ID: ______________________

1. Overall, I am satisfied with the use of **History Session Dialog** during the given tasks.

   1 2 3 4 5 6 7

   Strongly disagree Neutral Strongly agree

2. Overall, I am satisfied with the use of **History Interpretation Dialog** during the given tasks.

   1 2 3 4 5 6 7

   Strongly disagree Neutral Strongly agree

3. Overall, I am satisfied with the use of **Filmstrip Visualization** during the given tasks.

   1 2 3 4 5 6 7

   Strongly disagree Neutral Strongly agree

4. Overall, I am satisfied with **Visual Summary** during the given tasks.

   1 2 3 4 5 6 7

   Strongly disagree Neutral Strongly agree

5. Investigating **high-level interpretation** (grouping of events) was better to understand the history of the given documents than understanding low-level history events (e.g. add, delete, move and resize)

   1 2 3 4 5 6 7

   Strongly disagree Neutral Strongly agree

6. **<u>Keywords</u>** provided with grouping in **History Interpretation Dialog**, **Filmstrip Visualization** and **Visual Summary** were useful to understand the given documents throughout history.

   1 2 3 4 5 6 7

   Strongly disagree Neutral Strongly agree

7. Comparing to **History Session Dialog**, **<u>History Interpretation Dialog</u>** provides better understanding of the given document throughout history.

   1 2 3 4 5 6 7

   Strongly disagree Neutral Strongly agree

8. History filter in **<u>History Interpretation Dialog</u>** was helpful to narrow down the history navigation process.

   1 2 3 4 5 6 7

   Strongly disagree Neutral Strongly agree

9. **<u>The list of thumbnail captures</u>** in **Filmstrip Visualization** and **Visual Summary** was helpful to understand how the given document has been changed throughout history.

   1 2 3 4 5 6 7

   Strongly disagree Neutral Strongly agree

10. Having **<u>"detail textual summary of history"</u>** in **Filmstrip Visualization** was useful to navigate to the specific time spot in the history of the given document.

    1 2 3 4 5 6 7

    Strongly disagree Neutral Strongly agree

11. Having **<u>"detail visual summary of history"</u>** in **Visual Summary** was useful to navigate to the specific time spot in the history of the given document.

    1 2 3 4 5 6 7

    Strongly disagree Neutral Strongly agree

12. **Visual Summary**'s **visual hints** that provide the changes of a VKB workspace were helpful to quickly detect the changes throughout history.

1	2	3	4	5	6	7
Strongly disagree			Neutral			Strongly agree

13. **The textual and visual hints** that provide the contribution of authors were useful during the navigation of history via **History Interpretation Dialog, Filmstrip visualization, and Visual Summary**.

1	2	3	4	5	6	7
Strongly disagree			Neutral			Strongly agree

14. Which interface was the most useful to find target information / event in the history?

( ) **History Session Dialog**
( ) **History Interpretation Dialog**
( ) **Filmstrip Visualization**
( ) **Visual Summary**

15. Which interface was the most useful to orient yourself during history navigation?

( ) **History Session Dialog**
( ) **History Interpretation Dialog**
( ) **Filmstrip Visualization**
( ) **Visual Summary**

16. Which interfaces was the most intuitive for performing the given task?

( ) **History Session Dialog**
( ) **History Interpretation Dialog**
( ) **Filmstrip Visualization**
( ) **Visual Summary**

17. Which interfaces was the most difficult to use during the given task?

    ( ) **History Session Dialog**
    ( ) **History Interpretation Dialog**
    ( ) **Filmstrip Visualization**
    ( ) **Visual Summary**

18. Please rank the four interfaces in order of your preference. (1 ~ 4)

    ( ) **History Session Dialog**
    ( ) **History Interpretation Dialog**
    ( ) **Filmstrip Visualization**
    ( ) **Visual Summary**

    18.1. Please let us know why the ranked 4th interface is not preferable.

    18.2. Please let us know why the ranked 1st interface is preferable.

General comments:

____

____

____

____

____

VITA

Name: Do Hyoung Kim

Address: Department of Computer Science, Texas A&M University
TAMU 3112 College Station, TX 77843-3112

Email Address: kim@cse.tamu.edu

Education: B.S., School of Computing, Soongsil University, Korea, 1997
M.S., Computer Engineering, Seoul National University, Korea, 2002
Ph.D., Computer Science, Texas A&M University, 2011